




高等院校计算机科学与技术  
“十五”规划教材

# 多媒体技术及其应用



钟玉琢 沈 洪 吕小星 等编著



 机械工业出版社  
CHINA MACHINE PRESS

高等院校计算机科学与技术“十五”规划教材

# 多媒体技术及其应用

钟玉琢 沈 洪 吕小星 等编著

机械工业出版社

多媒体计算机技术及其产品是当今计算机产业发展的新领域。本书从设计、开发和应用的角度,综合讲述了多媒体计算机技术的原理和关键技术。主要内容有:多媒体计算机关键技术、现状及发展趋势;多媒体数据压缩编码技术;多媒体计算机的硬件和软件系统结构及其应用技术;超文本和超媒体;多媒体著作工具的使用、动画和视频的制作、多媒体数据库及多媒体视频会议系统等。

本书可作为普通高等院校相关专业多媒体计算机课程教材,可供大专院校师生阅读,也可供从事多媒体计算机研制、开发及应用人员学习、参考。

## 图书在版编目(CIP)数据

多媒体技术及其应用/钟玉琢等编著. —北京:机械工业出版社, 2003.6

高等院校计算机科学与技术“十五”规划教材

ISBN 7-111-12241-0

I. 多... II. 钟... III. 多媒体技术—高等院校—教材 IV. TP37

中国版本图书馆 CIP 数据核字(2003)第 039398 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 陈振虹

责任印制: 闫 焱

北京京丰印刷厂印刷·新华书店北京发行所发行

2003 年 6 月第 1 版·第 1 次印刷

787mm×1092mm

0001—5000 册

定价: 33.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010) 68993821、88379646

封面无防伪标均为盗版

## 出版说明

信息技术高度普及的今天，具备一定层次的信息技术素养成为社会素质教育的一个重要目标，由此对高等院校的计算机专业教育提出了更高更新的要求。教育水平提高的关键是教学质量，那么对教学质量有直接影响的教材建设就成为了计算机专业教育的根本，为重中之重。

适逢高等院校计算机专业教育改革的关键时期，为配合相关的教材建设，机械工业出版社会同全国在该领域内享誉盛名、具备雄厚师资和技术力量的高等院校，包括清华大学、上海交通大学、南京大学、成都电子科技大学、东南大学、西安电子科技大学、解放军理工大学、北京科技大学等重点名校，组织了多位长期从事教学工作的骨干教师，集思广益，对当前高等院校的教学现状开展了广泛而深入的研讨，继而紧密结合当前技术发展需要并针对教学改革所提出的问题，精心编写了这套面向普通高等院校计算机专业的系列教材，并陆续出版。

本套教材内容覆盖了普通高等院校计算机专业学生的必修课程，另外还恰如其分地添加了一些选修课程，总体上分为基础、软件、硬件、网络和多媒体五大类。在编写过程中，对教学改革力度比较大、内容新颖以及各院校急需的并且适应社会经济发展的新教材，优先选择出版。

本套教材注重系统性、普及性和实用性，力求达到专业基础课教材概念清晰、深度合理的标准，并且注意与专业课教学的衔接；专业课教材覆盖面广、深浅适中，在体现相关领域最新发展的同时注重理论联系实际。全套教材体现了教育改革的最新思想，可作为高等院校计算机科学与技术专业的教学用书，同时也是培训班和自学使用的最佳教材。

机械工业出版社



# 前 言

21 世纪的人类社会将是信息化社会,以信息技术为主要标志的高新技术产业在整个经济中的比重不断增长,多媒体技术及其产品是当今计算机产业发展的新领域和信息化的重要内容。世界上许多国家对多媒体技术的研究和应用都给予了极大的重视,并投入大量的人力、物力来开发先进的多媒体信息技术及相关产品,试图占领庞大的多媒体市场。

1994 年我国国家经贸委经过充分论证,将多媒体技术列入国家技术开发的重点项目计划,并在多媒体基础技术、多媒体平台及多媒体应用等方面给予了重点支持。几年来,我国在多媒体芯片和板级产品、CD 系列数字影碟机、多媒体电子光盘出版物、分布式多媒体信息系统、多媒体汉语语音交互技术、DVD 高密度数字光盘及多媒体通信计算机等方面有了长足的进步,并涌现了一批在多媒体领域效率较高的企业和相关的多媒体科技人才,从而成为计算机产业新的增长点。

1992 年由中国计算机学会和中国图像图形学会的多媒体专业委员会在清华大学联合召开了首届全国多媒体技术学术会议,以后在杭州浙江大学、上海大学、广州中山大学、武汉华中理工大学、西安交通大学、长沙国防科技大学、成都电子科技大学分别召开了第 2 到第 11 届全国多媒体技术学术会议。每届学术会议,大约都有 120 篇论文、将近 200 位代表参加。每次会议都对多媒体技术基础理论、多媒体应用支撑平台、分布式多媒体技术及多媒体应用等技术展开了广泛深入的学术交流和讨论。这些全国性的多媒体技术学术会议的召开,促进了国内外多媒体技术的交流和发展。

目前,国内很多高等院校纷纷为本科生开设多媒体计算机技术及其应用课程。为适应上述需求,应机械工业出版社邀请,特组织了北京联合大学沈洪副教授、北京广播电视大学吕小星副教授编写了这本《多媒体技术及其应用》教材,供普通高等院校计算机及相关专业本科生选用。

为了适应教学的要求,在编写中我们力求做到:

- 教学内容的系统性:根据教学规律,本教材内容由浅入深,注意理论联系实际,使学生既学好基础理论知识,又提高分析问题和解决问题的能力。
- 内容取材的新颖性和科学性:由于多媒体正在蓬勃发展,本教材注意了较新的科研成果、较新的标准、规范及其应用。

本书共分 9 章,第 1 章介绍多媒体计算机的定义、分类、现状和发展趋势;第 2 章和第 3 章介绍音频和视频信号的获取与处理;第 4 章阐述多媒体数据压缩编码算法及现行编码国际标准;第 5 章讲述多媒体计算机硬件及软件系统结构;第 6 章讲述超文本和超媒体;第 7 章介绍多媒体著作工具;第 8 章介绍多媒体数据库和基于内容检索;第 9 章介绍多媒体视频会议系统的开发和交互式电视技术。为了学生自学方便,在每章开头阐明本章要点,结尾给出本章小结和习题及相关的参考答案。

本书主要由钟玉琢教授、沈洪副教授和吕小星副教授编写。参加编写工作的还有蔡莲红教授、李树青教授、史元春副教授等。全书由钟玉琢教授审阅,在编写过程中,作者参考了

国内外有关多媒体计算机技术的书刊及文献资料，以及清华大学计算机系多媒体组的博士、硕士论文及科研报告。在整个编写过程中得到作者所在单位及同组其他成员的大力支持，在此表示衷心感谢。

多媒体计算机技术正处在蓬勃发展阶段，新的文献资料我们搜集的还不完整，限于作者学术水平，书中不足和错误之处，恳请读者给予批评指正。

钟玉琢

# 目 录

出版说明

前言

第 1 章 多媒体计算机技术概述 .....	1
1.1 多媒体计算机的定义和关键技术 .....	1
1.1.1 多媒体计算机的定义及其关键技术 .....	1
1.1.2 多媒体是计算机技术发展的必然趋势 .....	2
1.1.3 在多媒体计算机发展史上卓有成效的公司和系统 .....	3
1.2 多媒体技术促进了通信、娱乐和计算机的融合 .....	5
1.2.1 多媒体技术是解决常规电视数字化及高清晰度电视切实可行的方案 .....	6
1.2.2 用多媒体技术制作 DVD 及影视音响卡拉 OK 机 .....	8
1.2.3 数字家电网络平台 .....	9
1.3 多媒体计算机技术的发展和應用 .....	20
1.3.1 多媒体数据库 .....	20
1.3.2 多媒体通信 .....	21
1.3.3 多媒体创作工具及其应用 .....	22
1.3.4 多媒体计算机的发展趋势 .....	22
1.4 小结 .....	24
1.5 习题 .....	24
1.6 参考答案 .....	25
第 2 章 声卡和音频信息 .....	26
2.1 数字音频基础 .....	26
2.1.1 模拟音频和数字音频 .....	26
2.1.2 音频的数字化 .....	27
2.1.3 数字音频的文件格式 .....	28
2.1.4 音频信号的特点 .....	30
2.1.5 3D 音频 .....	31
2.2 声卡的组成与工作原理 .....	32
2.2.1 声卡的功能、技术指标与分类 .....	32
2.2.2 声卡的组成和布局 .....	36
2.2.3 声卡的工作原理 .....	39
2.2.4 SPDIF 数字音频接口 .....	41
2.2.5 音频卡的发展和改进 .....	43
2.3 音频编码基础和标准 .....	45
2.3.1 音频编码的基础 .....	45

2.3.2	音频编码标准介绍	47
2.4	小结	56
2.5	习题	56
2.6	参考答案	57
第3章	视频卡和视频信息处理	58
3.1	彩色空间表示及其转换	58
3.1.1	颜色的基本概念	59
3.1.2	彩色空间表示	60
3.1.3	彩色空间的转换及其实现技术	64
3.1.4	全电视信号	66
3.2	视频信息获取	67
3.2.1	视频信号获取器的工作原理	69
3.2.2	彩色全电视信号的数字锁相和数字译码	74
3.3	视频信号的采集	76
3.3.1	采集模拟视频	76
3.3.2	采集数字视频(VCD转换和HyperCam)	77
3.4	Premiere 简介	78
3.4.1	功能与特色	78
3.4.2	过渡效果	84
3.4.3	字幕和叠加	85
3.4.4	为视频添加滤镜效果	87
3.4.5	运动效果	89
3.4.6	生成电影	90
3.5	小结	90
3.6	习题	91
3.7	参考答案	92
第4章	多媒体数据压缩编码技术	93
4.1	多媒体数据压缩的重要性和分类	93
4.1.1	多媒体数据压缩的必要性	93
4.1.2	多媒体数据压缩的可能性	94
4.1.3	多媒体数据压缩方法的分类	95
4.2	预测编码	97
4.2.1	预测编码的基本原理	98
4.2.2	自适应预测编码	101
4.3	变换编码	102
4.3.1	变换编码的基本原理综述	102
4.3.2	离散余弦变换——DCT变换	104
4.4	统计编码	108
4.4.1	统计编码原理——信息量和信息熵	108

4.4.2	哈夫曼编码 .....	109
4.4.3	算术编码 .....	111
4.5	多媒体数据压缩编码的国际标准 .....	118
4.5.1	JPEG 标准 .....	118
4.5.2	MPEG-1 标准 .....	130
4.5.3	MPEG-2 标准 .....	135
4.5.4	MPEG-4 标准 .....	137
4.5.5	MPEG-7 标准 .....	150
4.5.6	MPEG-21 标准 .....	152
4.6	小结 .....	156
4.7	习题 .....	156
4.8	参考答案 .....	157
第 5 章	多媒体计算机硬件及软件系统结构 .....	159
5.1	光盘交互式多媒体计算机系统——CD-I .....	160
5.1.1	CD-I 系统的硬件结构 .....	160
5.1.2	CD-I 光盘实时操作系统 .....	172
5.2	数字视频交互式多媒体计算机系统——DVI .....	176
5.2.1	DVI 系统中的视频音频引擎 (AVE) .....	176
5.2.2	DVI 软件系统中的 AVSS .....	184
5.2.3	Windows 系统环境下开发的 AVK .....	189
5.3	将多媒体和通信功能集成到 CPU 芯片中 .....	196
5.3.1	集成的设计原则 .....	196
5.3.2	多媒体处理器——Mpact 和 Trimedia .....	198
5.3.3	Phenix 芯片和 MMX 技术 .....	215
5.4	小结 .....	232
5.5	习题 .....	232
5.6	参考答案 .....	234
第 6 章	超文本和超媒体 .....	235
6.1	概述 .....	235
6.1.1	基本概念 .....	235
6.1.2	典型的超文本系统 .....	238
6.2	超文本和超媒体系统的组成和结构 .....	240
6.2.1	超文本和超媒体的组成要素 .....	240
6.2.2	超媒体和超文本系统结构模型 .....	245
6.3	超文本和超媒体的文献模型 .....	249
6.3.1	概述 .....	249
6.3.2	ODA 模型 .....	250
6.3.3	HyTime 模型 .....	252
6.4	超文本与超媒体存在的问题及发展前景 .....	255

6.4.1	超文本与超媒体存在的问题 .....	255
6.4.2	超文本与超媒体发展的前景 .....	255
6.5	小结 .....	256
6.6	习题 .....	256
6.7	参考答案 .....	257
<b>第 7 章</b>	<b>多媒体著作工具 .....</b>	<b>258</b>
7.1	多媒体著作工具 .....	258
7.1.1	多媒体著作工具的分类 .....	258
7.1.2	多媒体著作工具的功能要求 .....	260
7.1.3	编程与著作工具 .....	261
7.2	Authorware 的使用 .....	263
7.2.1	Authorware 简介 .....	263
7.2.2	图标的功能、使用和素材的灵活引用 .....	265
7.2.3	Authorware 的高级制作 .....	272
7.2.4	Authorware 文件的打包和发行 .....	274
7.3	小结 .....	275
<b>第 8 章</b>	<b>多媒体数据库及基于内容检索 .....</b>	<b>277</b>
8.1	多媒体数据库 .....	278
8.1.1	多媒体数据的存储问题 .....	278
8.1.2	多媒体数据的管理 .....	279
8.1.3	多媒体数据库体系结构 .....	284
8.2	多媒体数据库基于内容检索 .....	287
8.2.1	基于内容检索系统的体系结构 .....	287
8.2.2	基于内容检索的关键技术 .....	289
8.2.3	一个基于内容检索系统的设计和实现 .....	298
8.3	小结 .....	303
8.4	习题 .....	304
8.5	参考答案 .....	304
<b>第 9 章</b>	<b>多媒体会议系统的开发和交互式电视技术 .....</b>	<b>306</b>
9.1	多媒体会议系统的开发 .....	306
9.1.1	视频会议系统的结构及标准 .....	306
9.1.2	综合业务多媒体终端的设计和实现 .....	313
9.1.3	多点控制单元 MCU .....	323
9.1.4	视频会议系统的服务质量 (QoS) 及资源管理 .....	330
9.1.5	视频会议系统的安全保密 .....	337
9.2	多媒体交互式电视技术 .....	341
9.2.1	概述 .....	341
9.2.2	交互式电视系统的功能和结构 .....	345
9.2.3	视频服务器的结构和设计 .....	349

9.2.4 交互式电视机顶盒的结构和设计 .....	364
9.3 小结 .....	369
9.4 习题 .....	370
9.5 参考答案 .....	370

# 第1章 多媒体计算机技术概述

## 本章要点

- 多媒体计算机的定义、分类、多媒体计算机和普通计算机的不同点以及多媒体计算机要解决的关键技术。
- 多媒体技术促进了通信、娱乐和计算机的融合。特别是多媒体技术是解决高清晰度电视切实可行的方案，用多媒体技术制作 VCD、DVD 及影视音响卡拉 OK 机、数字家电网络平台。
- 多媒体计算机技术的应用和发展：多媒体数据库、多媒体通信和多媒体创作工具以及多媒体计算机的发展趋势。

多媒体技术使计算机具有综合处理声音、文字、图像和视频的能力，它以形象丰富的声、文、图信息和方便的交互性，极大的改善了人机界面，改变了使用计算机的方式，从而为计算机进入人类生活和生产的各个领域打开了方便之门，给人们的工作、生活和娱乐带来深刻的变化。

## 1.1 多媒体计算机的定义和关键技术

媒体（Medium）在计算机领域中有两种含义，一是指用以存储信息的实体，如磁带、磁盘、光盘和半导体存储器；一是指信息的载体，如数字、文字、声音、图形和图像。多媒体技术中的媒体是指后者。

人类感知信息的途径是：

（1）视觉。是人类感知信息最重要的途径，人类从外部世界获取信息的 70%~80% 是从视觉获得的。

（2）听觉。人类从外部世界获取信息的 10% 是从听觉获得的。

（3）嗅觉、味觉、触觉。通过嗅、味、触觉获得的信息量约占 10%。

### 1.1.1 多媒体计算机的定义及其关键技术

多媒体计算机（Multimedia Computer）的定义是：计算机综合处理多种媒体信息，包括文本、图形、图像、音频和视频，使多种信息建立逻辑连接，集成为一个系统并具有交互性。简单地说：计算机综合处理声、文、图信息；具有集成性和交互性。总之，多媒体计算机具有信息载体多样性、集成性和交互性。

要把一台普通的计算机变成多媒体计算机要解决的关键技术是：

（1）视频音频信号获取技术。

（2）多媒体数据压缩编码和解码技术。



(3) 视频音频数据的实时处理和特技。

(4) 视频音频数据的输出技术。

多媒体计算机的分类, 从开发和生产厂商以及应用的角度出发可以分成两大类:

一类是家电制造厂商研制的电视计算机 Teleputer, 是把 CPU 放到家电中, 通过编程控制管理电视机、音响, 有人称它为“灵巧”电视 (Smart TV)。

另一类是计算机制造厂商研制的计算机电视 (Compuvision), 采用微处理器作为 CPU, 其它设备还有 VGA 卡、CD-ROM、音响设备以及扩展的多窗口系统, 有人说它的发展方向是 TV-Killer。

### 1.1.2 多媒体是计算机技术发展的必然趋势

在计算机发展的初期, 人们只能用数值这种媒体承载信息。当时只能通过 0 和 1 两种符号表示信息, 即用纸带和卡片的有孔或无孔表示信息, 纸带机和卡片机是主要的输入输出设备。0 和 1 很不直观, 也不方便, 输入输出的内容很难理解, 而且容易出错, 出了错也不容易发现。这一时代是使用机器语言的时代, 因此计算机应用只能限于极少数计算机专业人员。

20 世纪 50 年代到 70 年代, 出现了高级程序设计语言, 开始用文字作为信息的载体, 人们可以用文字 (如英文) 编写源程序, 输入计算机, 计算机处理的结果也可以用文字表示输出。这样, 人与计算机交往就直观、容易得多, 计算机的应用也就扩大到具有一般文化程度的科技人员。这时的输入输出设备主要是打字机、键盘和显示终端。使用英文文字同计算机交往, 对于文化水平较低的人, 特别是非英语国家, 仍然是件困难的事情。

20 世纪 80 年代开始, 人们致力于研究将声音、图形和图像作为新的信息媒体输入输出计算机, 这将使计算机的应用更为直观、容易。1984 年 Apple 公司的 Macintosh 个人计算机首先引进了“位映射”的图形机理, 用户接口开始使用 Mouse 驱动的窗口技术和图符 (Windows and icon), 受到广大用户的欢迎。这使得文化水平较低的公众, 包括儿童在内都能使用计算机。由于 Apple 采取发展多媒体技术、扩大用户层的方针, 使得它在个人计算机市场上成为唯一能同 IBM 公司相抗衡的力量。今天, 国际上下述几项技术又有了突出的进展:

(1) 超大规模集成电路的密度增加了。

(2) 超大规模集成电路的速度增加了。

(3) CD-ROM 可作为低成本、大容量 PC 机的只读存储器 (可换的 5in 盘片, 每片容量为 600MB, 以及 DVD (单面 4.7GB))。

(4) 双通道 VRAM、RDRAM 的引进。

(5) 网络技术的广泛使用。

这 5 项计算机基本技术的进展, 有效地带动了数字视频压缩算法和视频处理器结构的改进, 促使 10 年前单色文本 / 图形子系统转变成今天的彩色丰富、高清晰度显示子系统, 同时能够做到全屏幕、全运动的视频图像, 高清晰度的静态图像, 视频特技, 三维实时的全电视信号以及高速真彩色图形, 同时还有高保真度的音响信息。

综上所述, 无论从半导体的发展还是从计算机进步的角度, 或者从普及计算机应用、拓宽计算机处理信息类型看, 利用多媒体是计算机技术发展的必然趋势。

和 AVK 最主要的任务是为音频和视频数据流相关同步提供需要的实时任务调度，实时的数据压缩和解压缩，实时地复制和改变比例尺，建立位映射，管理控制它们将其送至显示缓冲区等。

### 3. Commodore 公司的 Amiga 系统

Commodore 公司在 1985 年率先在世界上推出了第一个多媒体计算机系统 Amiga。在 1989 年秋美国的 Comdex 博览会上，Commodore 公司展示了 Amiga 系统一个完整的系列。当时，该公司已推出了 Amiga 500, 1000, 1500, 2000, 2500 以及 3000 等型号的产品，它们可分别配置 Motorola 公司生产的 68000, 68020 以及 68030 不同型号的 CPU 以及不同容量的 RAM。为了提高视频和音响信息的处理速度，Commodore 公司在 Amiga 系统中采用了三个专用芯片：Agnus (8370)，Paula (8364) 以及 Denise (8362)。

Amiga 系统的结构与 68000 微机系统以及前面介绍的 CD-I 系统非常相似，只是在系统总线上连接了很有特色的三个专用芯片。下面简单介绍一下 3 个专用芯片的结构：

(1) Agnus (8370) 是专用的动画制作芯片，芯片中有 5 个 DMA 控制逻辑：视频 DMA，音频 DMA，位平面 DAM，软盘和刷新电路 DAM 以及位映射控制部件的 DMA 控制逻辑线路及其需要的控制寄存器，它们通过内部总线与专用芯片内部的图形协处理器连在一起。因为在 Agnus 有较多的控制寄存器，所以有寄存器地址译码器以及寄存器地址存储器译码器，此外还有系统总线的接口电路、缓冲器、多路开关以及时钟发生器等。

概括起来 Agnus 的功能是：

- 用硬件显示移动数据，允许高速的动画制作。
- 显示同步协处理器。
- 控制 25 个通道的 DMA，使 CPU 以最小的开销处理盘、声音和视频信息。
- 从 28MHz 振荡器产生系统时钟。
- 为视频 RAM (VRAM) 和扩展 RAM 卡提供所有的控制信号。
- 为 VRAM 和扩展 RAM 提供地址。

(2) Paula (8364) 是专用音响处理及外设接口芯片，芯片中音响处理器、盘控制器、异步通信接口以及电位计通道接口都连接到内部总线的设备译码器上。音响处理器是由 2 路数据寄存器、两个音响控制计数器及 4 路 D/A 变换器组成。它可以通过 DMA 的方式和 Amiga 系统的存储器以及其他设备交换音响信息，在 Paula 的音响处理器中处理音响信息，最后经过 D/A 变换器，可把 4 路两对立体声信号输出到音响设备中。盘控制逻辑也通过 DMA 的方式将 Amiga 系统中存储的数据通过盘控制器输出到盘上；反之也可将盘上数据通过盘控制器读入到 Amiga 系统中。此外，异步通信接口和电位计通道控制逻辑以 I/O 方式进行数据传输。该芯片的主要功能是输出 4 路两个立体声道、9 个八音阶，使用音频放大和频率调制，还有异步通信接口、盘控制器以及电位计通道接口。

(3) Denise (8362) 是专用的图形芯片，它有位平面数据寄存器、位平面控制以及位平面串行输出器；硬件游标数据寄存器、硬件游标串行连续化器以及位置比较逻辑；碰撞控制逻辑、碰撞检测逻辑以及碰撞存储逻辑；优先排队控制逻辑以及位平面排队和控制寄存器；彩色选择译码器以及 32 位彩色输出寄存器；Mouse 计数器。由上述可见，它就是多功能的彩色图形控制器，它可以控制不同分辨率的输出，从 320×200 像素到 640×400 像素；在电视机和 RGB 彩色监视器屏幕上可同时显示 4096 种颜色；有 8 个可重复使用的“硬件游标”控

制器。

Amiga 3000 型采用了 25MHz 的 68030 作为 CPU,配有协处理器,内存最大容量为 16MB, 9×100MB 硬磁盘以及任选 Ethernet, Novell Netware 和 UNIX 网络和软件。

为了适应不同用户对多媒体技术的需要,Commodore 公司提供了一个多任务 Amiga 操作系统,它有上下拉的菜单,多窗口,图符 (Icon) 以及 PM (Presentation Manager) 等功能。同时,配备了大量应用软件,如能绘制动画,制作电视片头及作曲等专用软件。该公司还推出了一个 Amiga Vision 多媒体的著作系统,为用户提供一个完备的图符编程语言 (A Complete Iconic Programming Language)。

#### 4. Apple 公司的 HyperCard

Apple 公司的 Macintosh 系统具有公认的良好图形特性,它是桌上出版和桌上展示系统的先驱。Apple 公司的多媒体系统也有人称之为桌上媒体,它实质上是把高质量的音响及活动的视频图像加到原来的 Macintosh 系统中,能够把上述特性连在一起的是 HyperCard 及其兼容软件。HyperCard 是以卡片 (Card) 为节点的超文本 (Hypertext) 系统,基本的信息单元是卡片或称节点,一个卡片可充满整个屏幕。一组卡片称为卡堆 (Stack),可以认为卡堆是 HyperCard 中的文件,同类和相关的卡片可在一个卡堆内。每个卡片不仅是字符,还包括图形、图像和声音。HyperCard 系统提供了许多命令或工具,通过鼠标或键盘实现控制完成卡片的浏览、编辑、制作,信息的输入、修改、检索。它能把简单的数据库、复杂的文本程序、编程语言及著作系统组成一个快速灵活的软件包。HyperCard 的数据库和所有的 MAC 的数据格式兼容,并开发有直接的连接电路、光扫描器以及 CD-ROM 驱动器连接。为了使 HyperCard 和这些外部设备相连接,Apple 公司已经公布了一个多媒体协议和驱动程序标准集,叫做 AMCA (Apple Media Control Architecture)。AMCA 是系统级的结构,用来访问视频光盘、音频光盘以及录像带的信息,软件工作人员不用为多媒体外部设备写专门的驱动程序。

Apple 公司原来选用 Mac SE 和 Mac II 作为多媒体计算机的平台,后来选用了 68030 微处理器作为 CPU,直接寻址最多可达 8MB,视频适配器板可在 16M 种不同颜色中同时显示其中 256 种颜色。音响媒体接口板和 HyperCard 软件兼容,能够提供良好的语音、音响效果,通过语音分析和识别能够代替键盘、鼠标以及操纵杆的功能。

为了快速、实时地处理视频和语音信号,Apple 公司和 MIT 的媒体实验室合作,组成新一代技术研究小组开发视频和音频信号压缩编码和解码技术。为了传输视频信号,他们提出了高速的宽带网以及对称的压缩编码和解码技术,并已研制出了这种样机。

## 1.2 多媒体技术促进了通信、娱乐和计算机的融合

所谓通信、娱乐和计算机的融合,即把消费类电子产品:电话、电视、图文传真机、音响、录像机与计算机融为一体,由计算机完成视频音频信号的采集,压缩和解压缩,实时处理视频和音频及其特技、视频的多窗口显示及音频的立体声输出,从而形成新一代的产品,为人类的的生活和工作提供全新的信息服务。

### 1.2.1 多媒体技术是解决常规电视数字化及高清晰度电视切实可行的方案

前几年,在美国成立了一个高级电视研究集团 (ARTC),它采用 MPEG 压缩编码标准,

同时播出方案，打包数据结构以及双层传输技术，比之早些时候日本推出的模拟式的高清晰度电视，这是一个切实可行的方案。

目前研制的 HDTV 有下述几个特点：

(1) 采用国际标准的压缩编码算法 MPEG-2，这意味着它能以 MPEG，JPEG 压缩编码算法为基础的多媒体计算机兼容，并与其互连通信。

(2) 采用打包数据结构，当电视信号在视频通道传输时，图像和声音数据分成不同分量，在大多数情况下，这些分量要遵循大小和次序的限制。HDTV 将图像和声音信息以及用于多媒体服务的附加数据以包的方式传送。这些数据可以是任意大小，只要它们符合频道特性，能以随机次序传送就可，这些数据包能够动态分配，使 HDTV 能与计算机、多媒体娱乐、教育系统及录像机通信，打开了将电视机、计算机和通信融为一体、通向更灵活服务领域的大门。

(3) 采用双层传输技术，双层传输技术保证 HDTV 的可靠性和抗干扰性。它将信息分开传送，最重要的数据放到具有高优先级的载波上传输，其它数据则放到具有标准优先级的载波上传输。

采用多媒体计算机技术制造 HDTV，它可以支持任意分辨率的输出，输入输出分辨率可以独立，输出分辨率可以任意变化，可以用任意窗口尺寸输出。同时，它还能赋予 HDTV 很多新的功能，如图形功能，视频音频特技以及交互式功能。

常规电视数字化技术及交互式电视技术（包括点播电视技术 VOD）都是当前世界上的热点课题，最佳的解决办法是采用数字式视频，数字式音频及 MPEG 压缩编码算法，以便于数据传输、存储及计算机控制和管理。世界上很多人的公司都在从事这方面的开发和研究，几年前，汤姆逊（Thomson）消费电子公司制定的战略目标是做常规电视数字化的先驱。具体做法是通过休斯银河（Hughes Galaxy）601 卫星开创世界首次全数字直接到户的卫星广播业务（DSS-Digital Satellites System 及 DBS-Direct Broadcast Service）。它能传送激光视盘和激光音盘的质量，使消费者很容易获得 120~150 个频道最受欢迎的电视节目。用户端只需要投资 600~800 美元，购置一个易于安装的 18in 或常规碟形天线，一个和录像机体积差不多的接收机/解码器以及一个易于控制和操作的遥控器。汤姆逊公司的 DSS 系统的销售目标是北美的每个家庭都选购一台，使他们有机会在家中观看卫星数字电视，这些新产品将改变用户娱乐、采购、学习甚至工作方式。我国中央电视台也开始向全世界播放常规电视数字化节目。

如何解决常规电视和高清晰度电视同播（Simulcast）问题，就像彩色电视采用 YUV 方案和黑白电视兼容的问题。最近推出的国际标准 MPEG-2 采用了分层的编码体系（Hierarchic Coding），提供了较好的可扩充性（Scalability）及互操作能力（Interoperability）。MPEG-2 整个视频比特流由逐级嵌入的若干层组成，这样不同复杂度的解码器可根据自身的能力从同一比特流中抽出不同层进行解码，得到不同质量、不同时间/空间分辨率的视频信号，分层编码使同一比特流能适应不同特性的解码器，极大地提高了系统的灵活性、有效性，同时也为视频通信系统向更高时间/空间分辨率过渡提供了技术保证。为了实现分层编码，MPEG-2 提供了四种工具：空间可扩展性（Spatial Scalability）、时间可扩展性（Temporal Scalability）、信噪比可扩展性（SNR Scalability）及数据分块（Data Partitioning）。为了支持灵活的性能价格比，MPEG-2 还提供了框架（Profile）与等级（Level）的概念，给出了丰富的编码方法，

灵活的操作模式以适合不同场合的需要。

最近几年的热点课题是交互式电视技术 (ITV)，因为交互式电视技术有较好的发展环境，较好的经济、社会效益及广阔的应用前景。从美国宣布“信息高速公路”计划后，全球掀起了信息高速公路的热潮，纷纷投资巨款建设国家信息基础设施 (NII)。我国也在积极、慎重地开展 CNII 计划，并于 1994 年 9 月正式建成开通了我国公用数字数据网 (CHINADDN)，它可为用户提供  $N \times 64\text{Kbit/s}$  ( $N=1 \sim 31$ )， $2.4 \sim 19.6\text{Kbit/s}$  数字专线业务，用户可以进行单向、双向及  $N$  向的广播、会议电视等点对点、点对多点以及多点对多点的传输业务；1993 年 9 月正式建成开通了我国公用分组交换数据网 (CHINAPAC)，该网已覆盖所有省会城市及地、县乡 2000 多个点，总容量已达 10 万多个；1995 年 6 月正式开通了我国公用计算机互联网 (CHINANET)，它是 CHINADDN 和 CHINAPAC 基础上的增值网。现在国内很多城市正在积极筹建 VOD (Video On Demand) 系统。

未来的信息高速公路上，传递最多的信息是交互式电视和其他视频信息，交互式电视有最广大的用户，潜在的用户量可以是几亿或数十亿。交互式电视用户可以坐在家里的机顶盒 (STB-Set Top Box) 前，通过单键遥控器和菜单，选择自己喜欢的电影、电视和新闻，它可以提供交互式电视教育、电视采购，视频游戏以及各种方便的电视、电话和数据信息服务。

交互式电视系统和分布式多媒体数据管理系统从机理上是完全一样的，交互式电视台把新闻和其他节目经过视频和音频的压缩存储到数据库中，用户可以利用机顶盒而不是多媒体工作站，通过网络点播各种广播节目。交互式电视最常用的是节目间的交互，即 VOD 系统，典型的 VOD 系统主要由下述四部分组成：

- 视频服务器。
- 编码器/路由器。
- 用户请求计算机和记账计算机。
- 机顶盒。

多媒体计算机技术在常规电视和高清晰度电视、影视节目制作中的应用可以分成两个层次：一是影视画面的制作，影视画面的生成可以采用计算机软件生成二维、三维动画画面，摄像机摄制真实的影视画面后采用数字图像处理技术制作影视画面，最后是采用计算机生成和实拍结合后用图像处理技术制作影视特技画面。这方面的成功示例已经很多：美国惊险科幻影片“侏罗纪公园” (Jurassic Park) 中史前动物恐龙的许多精彩镜头都是用计算机制作的，它荣获了奥斯卡最佳视觉效果奖；电影“刚果”、“真实的谎言”、“阿甘正传”及“狮子王”中很多画面都是用计算机制作的，它们都产生了极佳的影视效果。另一个层次是影视的后期制作，例如现在常用的数字式非线性编辑器，实质上是一台多媒体计算机，它需要有广播级质量的视频音频的获取和输出、压缩解压缩、实时处理和特技以及编辑功能。目前美国、加拿大、德国的一些公司研制生产了一些较好的产品在世界销售，例如：苹果机为平台的非线性编辑器以 Avid 产品占主流；Truevision 公司的 TARGA2000 是非线性编辑器的核心板卡；此外还有 Miro、Ops 以及 Fast 等公司的产品。非线性编辑器长远的发展方向，一方面是提高质量、速度，降低成本，更重要的发展是网络化的编辑器，它的好处是多个编辑器协同工作，还可以连通“素材库”、“节目库”和“播放库”，可以大大提高编辑的效率和质量。

1.2.2 用多媒体技术制作 DVD 及影视音响卡拉 OK 机

多媒体数据压缩和解压缩技术是多媒体计算机系统的关键技术，全世界很多半导体厂商都在积极开发、生产、销售图像压缩和解压缩芯片。美国 C-Cube 公司从 1989 年成立以来，已投资数千万美元，先后开发生产了 CL-550、CL-560 及 CL-460。它们是 JPEG 静态图像压缩和解压缩处理器以及 MPEG-1 解码器，同时还提供板级产品及开发系统。经过几年的努力宣传和推销，虽然连续几年在 Comdex 博览会上深受观众的欢迎和青睐，但该公司的产品在多媒体计算机领域中的销量欠佳。为了打开芯片的市场销路，该公司的董事长决定把芯片用到 VCD 的播放机中。

VCD 是 JVC、Philips、Matsushita 及 Sony 联合制定的数字电视视盘技术标准，它于 1993 年问世。我国安徽合肥万燕（WYai）公司（与美国、韩国合资）在世界上首先利用 MPEG 国际标准和 CD 光盘技术，研制了全功能影视音响卡拉 OK 机 CDK-320。

CDK-320 采用 VCD 的标准，一片 VCD 盘片可以存放 70min 影视节目，利用 MPEG-1 的音频编码技术将声音压缩到原来的 1/6，再利用帧内和帧间动态图像压缩技术使图像的分辨率可达：NTSC 制为  $352 \times 240 \times 30$ ，PAL 制为  $352 \times 288 \times 25$ ；声音质量可达 CD-DA 的质量。

VCD 播放系统主要有下述两种产品：

(1) MPEG 播放卡。最简单的 MPEG 播放卡是由 Mediamatics 公司研制的基于 Mx501™ 芯片的 MPEG 播放卡，全卡只有三块芯片，Mx501，4Mbit DRAM 及 40MHZOSC。

(2) VCD 播放机。由 CD-ROM 驱动器、MPEG 解压卡及控制操作电路组成。

MPEG 播放卡和 VCD 播放机的工作原理如图 1-1 到图 1-3 所示。图 1-1 是 Mediamatics 公司研制的 MX501 的结构框图：它由视频音频解码加速器、视频音频同步时基、DRAM 控制器和仲裁器、PCI 总线/主机寄存器组成。它在计算机系统中如图 1-2 所示。VCD 播放机的原理框图如图 1-3 所示，ES3204 是美国 ESS 公司设计制造的 MPEG-1 视频音频解码芯片。无论是 MPEG 播放卡，还是 VCD 播放机，它们都是从 CD-ROM 驱动器上的 CD 盘上读出串行的 MPEG 数据流信号及其他的控制信号，经过 MPEG 音频译码器、解出立体声的音频信号，再经过 A/D 变换器，通过卡拉 OK 处理器可以接收话筒输入的卡拉 OK 信号，经过混合叠加处理，放大输出到音响设备或家用电视机。另一路 MPEG 视频流，经过视频解码器、视频 D/A 变换器、NTSC/PAL 编码器形成全电视信号送到家用电视机，或者不经过 NTSC/PAL 编码器直接输出 RGB 信号到监视器。其他的还有控制信号，可以接收计算机通过键盘和鼠标发出的各种控制命令，对于 VCD 播放机可接收红外遥控器或面板控制按钮，控制播放、快进、快倒、暂停以及控制色彩亮度和音量等。下一步应该是 DVD 播放卡和 DVD 播放机，图 1-4 给出以 ESS 公司研制的 ES-3309 芯片为核心的 DVD 播放机的原理框图，它的工作原理与 VCD 播放机一样，只是视频和音频编解码标准不是 MPEG-1，而是 MPEG-2 或 AC-3。

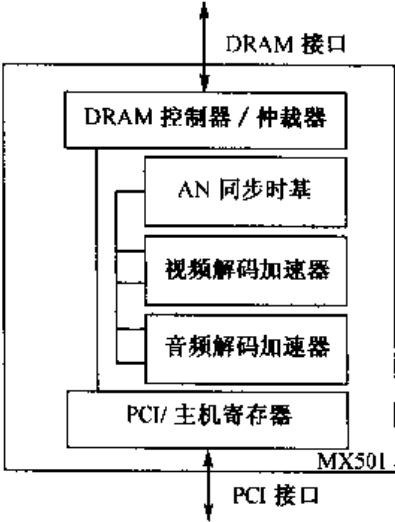


图 1-1 MX501 原理方框图

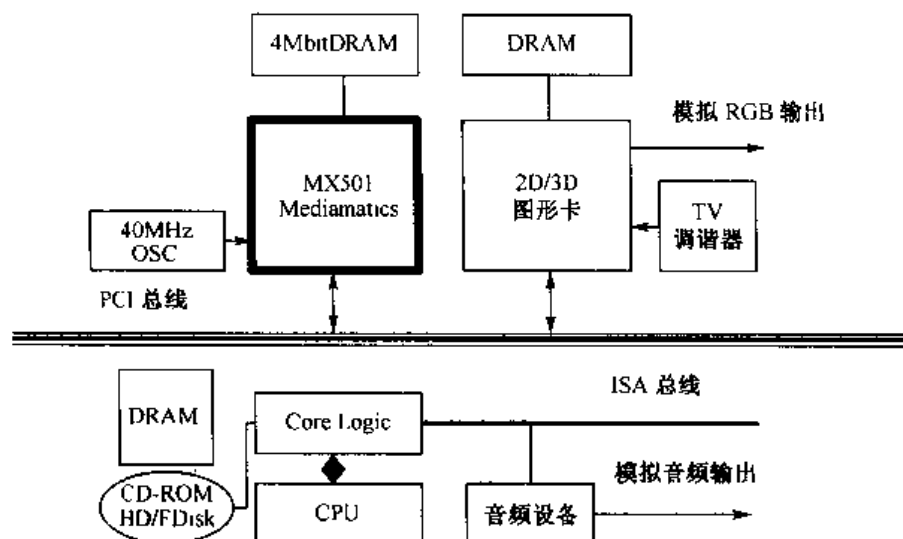


图 1-2 基于 MX501 MPEG-1 播放卡系统工作原理图

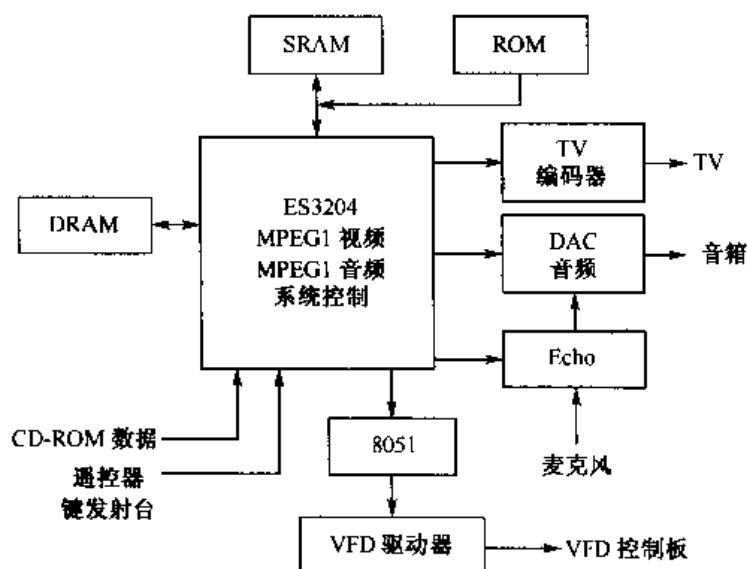


图 1-3 基于 ES-3204 芯片的 VCD 系统框图

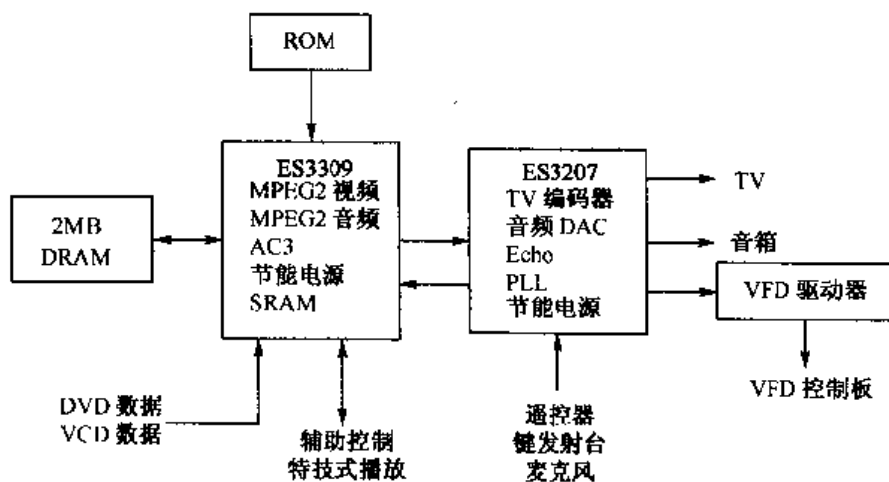


图 1-4 基于 ES-3309 芯片的 DVD 系统框图

### 1.2.3 数字家电网络平台

计算机技术进入后 PC 时代，无处不在的网，无处不在的计算（everything connecting, everywhere computing），成为当今计算机技术的潮流，同时带来家居环境的变革，PC 机、PDA 以及数字相机、蜂窝电话、数字电视等各种数字式消费电子产品开始涌入家庭，而浩瀚的网上信息更使人们把 Internet 访问、电子邮件、个人商务以及网上购物等作为购买电脑的主要目的，为了在家中更方便地获取信息、通信或娱乐，人们开始将所有这些信息家电连接成无线或有线的网络，同时给日常生活模式带来巨大的改变。

数字家电网络平台技术是两种研发群体的汇聚，一是多媒体计算机研究人员：他们从综合业务多媒体终端出发，在传统的 VOP（Video on demand）、视频会议、远程监控等处理图文信息多媒体技术上，开发适应家庭服务的数字家电网络平台。另一种是数字电视研发人员，他们从高清晰度电视的应用出发，在接收数字电视的多媒体处理基础上增强网络服务等其它功能以达到数字家电网络控制的目的。

#### 1. 数字家电网络平台的功能及硬软件结构

##### （1）数字家电网络平台的功能

数字家电网络平台的发展，应当以适合家庭应用环境的多功能集成为方向，以简便易用，安全可靠，价格低为设计基础，应当具有的功能有：

1) 接收并播放数字视频节目。包括 cable，陆地，卫星等连接方式的数字视频广播，即接收有线数字电视的 64QAM 调制编码信号，地面数字电视广播的 COFDM 调制信号和数字卫星 QPSK 调制信号，使用 MPEG-2 MP@ML 标准的数字音视频解码，同时输出复合视频信号和 S-VIDEO 信号，支持 MPEG-2 双声道立体声音频输出或 AC3 杜比 6 声道输出。

2) 支持多协议的 Internet 功能。具有良好的 Web 性能，可支持 TCP/IP、HTML、XML、HTTP、DNS、LDAP 等网络协议，使家电设备实现良好的 Internet 共享，包括网络内部及外部的数据共享。

3) 支持家电网络协议的家庭网络控制中心功能。即支持网络内部各种设备之间的数据通信，实时多媒体数据传输，多媒体交互式操作，即插即用（即家庭网络中的设备必须自动识别、发现、自我构造），网络之间也必须对等发现并列举出共享的设备以及服务，提供足够的宽带通信服务接口，如 ADSL、线缆调制器、卫星通信等，网络传输介质的兼容性与上层控制软件无关，可支持 BLUETOOTH，HOME PNA（电话线）、PLC（电源线）、HOME RF（无线射频）、IEEE1394、10/100BASET 红外等网络媒介，同时还应当能够对家用电器实施各种有效的控制机制。具有丰富的用户界面和一定的图形控制功能，使用户方便、直观地操作和控制各项应用，同时显示清晰、生动的图形界面，如画中画、多图像叠加等。

4) 具有家庭信息服务器的功能。为家庭事务管理提供必要的数据服务，最终完成家庭购物，家庭办公，家庭医疗，交互教学，交互游戏，视频邮件，视频点播，会议信息等全方位应用。

##### （2）数字家电网络平台硬件结构描述

根据数字家电网络平台的功能，可分为控制子系统，数字处理子系统，接口子系统，用户/扩展接口子系统，我们以 Philips Nexperia 数字电视平台（如图 1-5 所示）为例，介绍数字家电网络平台的硬件结构框架。



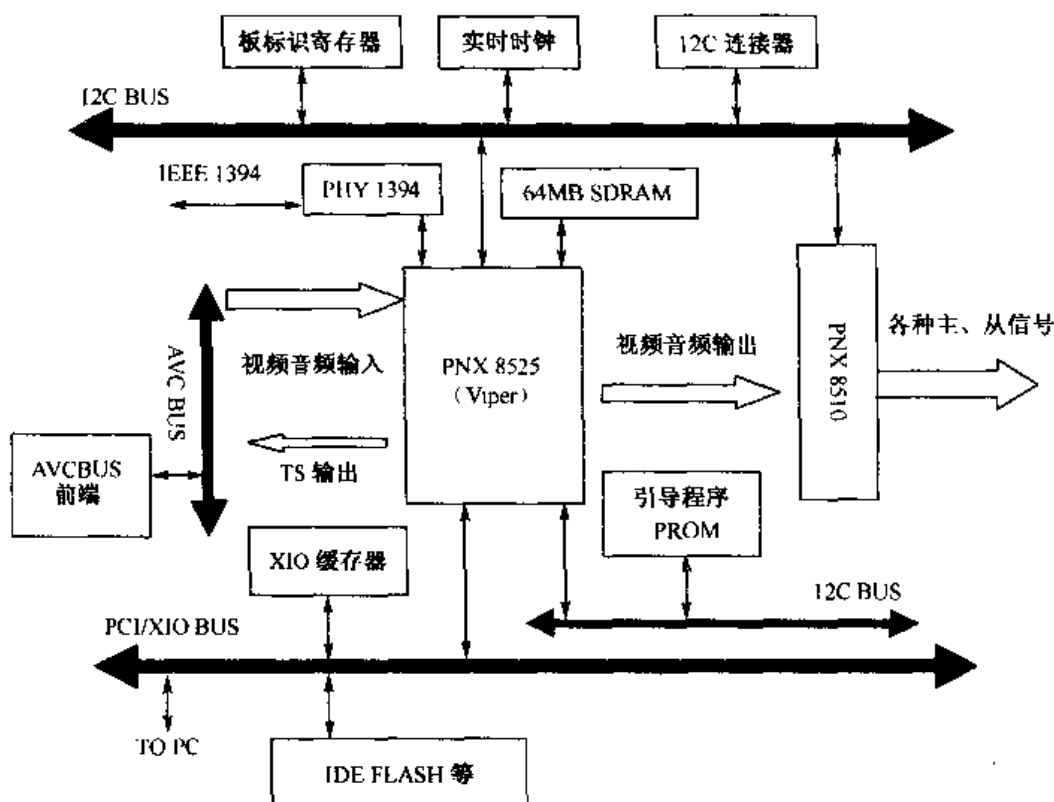


图 1-5 Philips Nexperia 数字家电网络平台硬件结构框图

Philips 的 Nexperia 平台为数字家电网络平台专门设计，为高清晰数字电视和高级机顶盒提供高性能的解决方案。Philips 公司研制开发了一个 PNx8500 系列芯片，它集成了两个可编程的处理器，一个用于控制功能的 32 位 MIPS RISC CPU，另一个专门用于多媒体处理的 TM-32 处理器。

#### 1) Nexperia 平台的特点:

- 丰富的硬件资源。Nexperia 的硬件模块和外围设备为 Internet 浏览、数字视频解码甚至是可视电话等应用提供了集中的支持。这些模块和外设包括视/音频解码器（特别是 MPEG-2 解码和解复用器）、图像增强模块、2D/3D 加速器以及 SPDIF、IEEE1394、USB、smartcard、PCI、MMI、UART 和 I2C 等接口。
- 多功能的总线系统。Nexperia 平台包括三个层次的内部总线，以提供最大的设计灵活性和速度。它们是：内存总线——提供高带宽的视频数据通路；PI 总线——用于 DMA；DVP-level2——用于将硬件从内存技术中抽象出来。
- 高性能的多处理器环境。Nexperia 总是使应用被分配到最合适的处理器上运行。它的设计把控制和实时的多媒体处理分离开来，从而在保证最短的响应时间的同时并不会影响视音频的处理，避免了图像冻结、丢帧等情况的发生。
- 完善的软件体系。Nexperia 完整的软件体系支持 Java 虚拟机和多种操作系统（包括 PSOS 和 WINCE）。硬件模块的驱动程序都是与操作系统无关的，可以很容易地在当前的操作系统上使用。作为扩展，Nexperia 还提供了一系列的流化软件模块包括 AC3、MPEG、MJPEG、TCP/IP/PPP 和语音识别等。

2) Nexperia 硬件平台——PNX8525 芯片。Nexperia 硬件平台的核心使用的是一块功能强大、高度集成的多媒体处理芯片 PNX8525，它包括输入模块、输出模块和处理模块。多

媒体数据会先被送到输入模块，然后由输入模块写入内存。处理模块从内存读出数据进行处理然后把结果回写。最后，输出模块从内存获得数据并输出到片外。

(a) 输入模块。MSP (MPEG System Processors) 是 MPEG 传输流数据在 PNX8525 芯片的主要入口点。PNX8525 有一个 MSP 部件，能够从不同的来源 (包括 IEEE1394 接口, DV2 和 DV3 输入, 以及 TSDMA) 同时接收两组不同的传输流数据。MSP 能够进行 MPEG-2 和 DIRECTV 传输流数据的解码, 包括解绕码, 解复用和利用 DMA 方式进行相应数据内存的存储。

VIP (Video Input Processor) 可以接收两个同时输入的经过 Philips 711x 系列视频解码器数字化的模拟视频信号。输入的视频流可以存储在主存储器中, 以帧的方式或是以场的方式。它的输出可以是 RGB 或是 YUV4: 2: 2 模式。VIP 支持的主要视频流速度为 13.5Mpix/s, 更高的输入速度最高可达到 40Mpix/s (在特殊场合下使用, 而且要限制在系统的带宽允许下)。

AI 模块包括两个 I2S 立体声入口, 可以和两个模拟视频输入流配合使用, 在 32 位下的最大采样频率为 96kHz。

SPDI 模块具有 SPDIF 输入, 由 TM32 处理器编程控制, 可以输入多种格式的音频数据, 如 IEC-1937 标准, AC-3 音频数据和 PCM 立体声。

(b) 输出模块。AICP (Advanced Image Composition Processors) 部件完成进一步的图像合成操作。系统提供两个 AICP 部件同时进行图像的合成操作, 并把合成后的图像分别送到 PNX8510 的两个显示通道上显示输出。根据 AICP 的模式选择, 可以输出 RGB 格式或 YUV 格式。

AO 模块提供两个面向 PNX8510 的立体声 I2S 输出。

PNX8510 提供最后阶段的视频和音频模拟化输出。两个 DENCs 支持 NTSC, PAL, SECAM 输出。六个 10 位的数模转换器组成了两个视频输出频道。第一个频道由四个 DAC 组成支持 SCART 输出。第二个由两个 DAC 组成支持 S-Video 输出。此外, PNX8510 还用四个 DAC 提供两个立体声声道输出。

SPDO 提供 SPDIF 输出。

(c) 处理模块。TM32 Core CPU 的主要技术指标如下:

- 0.18 $\mu$ m 的主频为 200MHz 的超长指令字的处理器。
- 每个时钟周期运行 5 条指令, 并带有媒体指令扩展。
- TM32 支持 32 位整型和 32 位 IEEE 兼容的浮点型数据结构, 支持 SIMD 操作。
- 能够达到浮点运算 1G 次/s, 16 位数据乘-加运算 800M 次/s 的计算能力。
- 具有片上的 16KB 的 8 路组相连的数据缓存和 32KB 的 8 路组相连的指令缓存。
- TM32 的数据带宽达到 1.6GB/s, 指令速率达到 5.6GB/s。

它的主要功能是在 PNX8525 中负责媒体和实时处理。如 slice 层次上的 MPEG 解码任务, 无法用片上硬件实现的视频处理算法和全部的音频处理功能。在上面运行 pSoS For Trimedia 操作系统。

PR3940 MIPS RISC Core 的主要技术指标如下:

- 主频 150MHz 的通用处理器。
- 内置 R4000 的基于 TLB 的 MMU (Memory Management Unit)。
- 分离的 16KB 2 路组相连指令缓存和 8KB 的 4 路组相连数据缓存。

它在 PNX8525 中主要负责运行操作系统和控制功能。对各个功能部件，进程，存储等的管理，控制 I/O 和 PCI 总线等数据交互。在上面运行 WinCE、pSoS for MIPS 操作系统。

Drawing Engine (2D) 引擎提供高速的高质量图像处理，包括静态区域填充，图像位复制，划线和单色位图扩展等，支持每像素 8 位、16 位和 32 位。

MBS (Memory-Based Scaler) 模块对于扩展操作提供加速，它将图像从内存中读出，进行相应的扩展操作后，再存到内存里。MBS 模块的最大优势是将图像的扩展部分同图像的显示部分分离，使得 MBS 可以在系统显示一幅图像的同时能够完成多幅图像的扩展工作。

VMPG (HL MPEG2 Decoder) 部件提供 HL MPEG2 在 slice 层次以下各种算法的硬件解码，支持所有的 18 种 ATSC 制式解码，对于高质量制式解码，VMPG 能够提供水平方向上二分之一的数据压缩，减少解码后图像占用的空间。

TM32CPU 上的软件可以通过 TSDMA 模块产生一个任意的传输流。这个传输流的包长度、时钟和包间隔都是可编程的。

DMA 用于将数据在系统的物理地址范围内任意的移动（在 DRAM 之间，XIO 和 PCI 之间）。

(3) 数字家电网络平台软件体系

数字家电网络平台的软件总体结构可分为：设备驱动层，基本操作系统层，逻辑资源层，中间件运行环境层和应用层。如表 1-1 所示。

表 1-1 数字家电网络平台软件体系

嵌入式应用	网络浏览	E Mail	游戏	传真	VOD	IP 电话	DVB
中间件和运行环境	统一的应用程序编程界面						
	资源管理器						
	HAVi 管理器	Jini	UPnP	Java TV	OPEN TV	HomeAPI	
	Java VM						
逻辑资源	TCP/IP	文件系统	图形界面		中文环境		
基本操作系统	Linux, Psos, OS Open 等嵌入式操作系统						
驱动程序	驱动程序管理器						
	HomePNA 驱动器		Bluetooth	IEEE1394	IEEE802.2	IEEE802.11	
	USB		CEBUS	POWER Line	SWAP	其它驱动	

1) 设备驱动程序层。就是对信息家电平台根据不同操作系统的设备驱动规则，为应用程序提供透明的、规范的控制机制所采用的软件接口，其实现方式一般由操作系统的核心态完成，一方面完成相应设备控制寄存器、中断的设置，另一方面完成设备的数据处理和存储空间管理。不同的处理器及硬件结构，设备驱动程序的编制会有差别，尤其是信息家电平台的中央处理器厂家众多，各种平台的结构存在差异，给编程带来较大难度。在 IBM REDWOODIII 平台上的设备驱动既有块设备 RS232, OSD 等，又有字符设备 IRDA，还有基于解复用，解码等由其它 DSP 独立处理的设备驱动模式。

2) 基本操作系统层。目前的嵌入式操作系统有 Linux, PSOS, Osopen, WIN\_CE, QAX 等，嵌入式操作系统应当完成普通 OS 的大部分功能如进程管理，线程管理，内存管理，设备管理，文件系统等，如果操作系统的设备驱动是基于文件系统实施的话，文件系统部分就不可缺少。基本操作系统和微内核的区别在于前者本身就是一个完整的操作系统，可以直接

为应用程序和上层软件提供核心服务，应用程序可直接调用其功能，基本操作系统层的调用界面遵循 POSIX 系列标准，以保证应用程序的良好可移植性。

3) 逻辑资源层。处于这一层的软件均为可选择安装的模块，可以根据特定的环境来选用，这一层中常用的模块有以下几种。

- 文件系统。在信息家电平台中，将文件系统从基本操作系统中分离出来，作为一个单独的模块，这样就给用户较大的选择，用户可以根据兼容性或其它方面的需要选择不同格式的文件系统，如 FAT，FAT32，NTFS，CDFS，I 节点等等。
- TCP/IP 模块。TCP/IP 协议是计算机通信的工业标准，是访问 Internet 的基础，也是管理 Java 设备的前提，该模块一般是必须的。
- 界面系统。信息家电平台的硬件平台不同，显示设备也大不相同，它们可以是液晶的字符显示板或者高分辨率的彩色显示器，有的甚至不配备显示器，针对这些情况，信息家电平台的界面系统也应该是多种形式的，提供图形窗口系统，字符窗口系统，行字符系统等。
- 中文环境。中文环境主要用来与界面系统配合使用，主要完成中文的输入、处理和显示，一般包括汉化内核，移植现有的中文字库，如点阵 12×13，16×16 和 24×24 等，移植现有的中文输入法如笔顺输入法和其它输入法，汉化重要的操作系统的运行库。

4) 中间件/运行环境层。介于应用和操作系统平台之间，为虚拟机的方式，为应用程序提供一个相对统一的开发环境和运行环境，便于应用程序的开发移植，一般插入 Pantalk，Java，HTML，MHEG5 等的解释器，同时可管理各种厂商制定的连接网络，并将它们映射到一个统一的管理网中。

在连接网络的规范中，包括三个层次：控制接口，控制协议和组织结构。

中间件技术是数字家电网络平台软件的关键环节，目前的 HAVi，Jini，UPnP，Java TV 等家庭网络方案的技术实质，就是基于中间件概念来完成应用功能的。目前的 OSGI (open service gateway initiative) 服务网关就是实施各种设备网络如 Jini 和 HAVi 的互连和管理，集成了全部或部分已存在的产品如 (数字或模拟) 数字家电网络平台，线缆调制解调器，集线器，家用网关，报警系统，能源管理系统，消费电子，个人计算机等的控制机制。这种服务网关将采用已有的 Java 标准如 Jini，并集成其他非 Java 标准的其他家庭网络协议，将这些设备连接到中心管理系统，为服务商提供网关和服务的接入。

应用层的概念比较清晰，目前的应用会以 Java TV 或其它基于 Java 的应用开发平台为主，其它非 Java 平台的厂商也正在进行 Java 应用的嵌入，以便于实现众多的网络应用。

## 2. 数字家电网络平台外部联网方案

将家庭内部网络与户外现存的各种网络互联可以采用有线或无线方式，有线方式有传统的 Modem，目前的 Cable Modem，xDSL 技术，数据广播技术等，无线的方式有 WAP，LDMS 等。传统 Modem 的最大传输速率为 56Kbit/s，采用的标准有 X2，V.90 等，ISDN 作为窄带数据交换，这些仍是目前常用的方式。

Cable Modem 方式是数字家电网络平台外向联网的最具潜力的方式，主要采用 QAM 信道调制方式，其中 64QAM，下行数据传输为 27Mbit/s，256QAM 下行数据传输为 36Mbit/s，16QAM 下行数据传输为 10Mbit/s，Cable Modem 有两种标准体系：IEEE802.14 和 MCNS—DOCSIS，前者的物理层支持 IEEE：ITU Annex A，B 和 C，64/256 QAM，介质访问子层支

持 ATM; 后者 MCNS-DOCSIS 的物理层支持 ANNEX B, 协议访问层支持变长数据包机制。Cable Modem 的接口部分包括: 10/100 Base-T Ethernet 卡, 外部通用串行总线 Modem。

xDSL 技术是通过传统电话线 POTS 铜线实现高速的数据传输, 并且数据和语音在同一条线上传输, 其工作频率大致在 100kHz, 传输速率可以从 2Mbit/s 到 30Mbit/s, 但是一般低于 10Mbit/s, 通常通过 Modem 上传数据, 现今此技术存在许多标准。它们有 ADSL, Lite, CDSL, HDSL 和 IDSL 等, ADSL 和 HDSL 使用得较多, 其他的还有 RDSL (自适应用户数字线), 它可根据传输线的质量和传输距离动态地调整访问速度, VDSL (甚高速用户数字线) 是传输速率最快的一种, 在一对双绞线上可支持上行传输速率 13~52Mbit/s 和下行传输速率 1.5~2.3Mbit/s, 但传输距离只有 300~1400m, 可以支持高清晰度电视 (HDTV) 信号。

利用无线技术进行家庭网络接入, 上行通道多采用模拟调制解调器, 通过标准 SLIP / PPP 连接到本地 ISP。实际速度根据调制解调器的速率而定。其中局域性多点分布服务 (LMDS) 是一种双向的数字式广播系统, 主要利用地面设备进行数据收发。LMDS 使用 G 波段, 由于频率高, 所以采用集线站和网络中心的转发器传输数据。这个通信中心的作用是处理所有的路径选择、线路交换以及桥接到 Internet 上等一些列问题。虽然 LMDS 不是一个完全的交换式网络, 但是它可以建立虚拟的点对点连接。正如其他高速访问方案一样, LMDS 也是非对称的系统。

### 3. 家庭内部网络协议层方案及标准

HAVi, Jini, UPnP 是目前解决家庭内部网络的较有影响的协议层规范, 世界许多著名 IT 业厂商已加入到这几个标准中, 对今后数字家电网络平台的整体应用框架有较大影响。

#### (1) HAVi 体系

HAVi (Home Audio/Video Interoperability), 是关于家庭网络中音频/视频电子产品的互联和控制方面的标准, 该标准建立在 IEEE-1394 的底层协议基础上, 主要实现 HAVi 设备之间的数字音频/视频内容的传送以及对该内容的操作, 如播放, 录像, 回放。典型的 AV 内容有由信息家电平台接收的数字电视和由数字录像机、数字相机、CD 或 MD 产生的内容。互操作性是 HAVi 标准的主要特点, 一个 HAVi 设备上的应用软件可以探测到并使用连入 HAVi 网络上的其他设备提供的功能。

HAVi 的软件基础包括系统软件基础和设备控制模块 (DCMS), 前者包括信息机制、登录机制、事件驱动机制、资源管理机制、信息流管理机制和 DCM 管理机制; 后者控制 HAVi 设备的特定功能, 比如 VCR 或相机。DCMS 和应用软件都是可安装的, 在一个 HAVi 网络中它们是否存在取决于相应的设备是否被安装。每种应用软件都有符合 HAVi 规范的 APIS, 通过传递消息, 一个 HAVi 设备的软件可以成为另一个软件的 APIS, HAVi 消息机制可以确保消息正确地传递给目标软件, 目标软件根据 APIS 的定义执行对应的功能。如果一个 HAVi 设备的软件在另一个设备上, 消息机制可以负责将消息通过 HAVi 网络传递给那个设备上的对等消息机制, 然后由其转发给目标软件, 该过程对应用软件来说是透明的。几个应用软件可以同时 HAVi 设备上执行。HAVi 定义了一个强大的资源管理机制, 它可以在几个不同的应用软件试图控制同一个 DCM 时处理可能发生的冲突, 资源管理机构允许共享设备, 能否共享取决于设备本身。

HAVi 的信息管理机制负责对网络中实时 AV 数据流的传送进行初始化和终止运行, 它依赖于其它的标准如 IEC 61883 来定义 HAVi 网络中传送这些内容的方式。HAVi 还定

全部成功，要么这些操作全部取消。

### (3) UPnP 技术

UPnP (Universal Plug and Play) 扩展了 PNP 简单自动的原则，通过使用开放式的硬件软件标准和协议，实现家电设备网络控制。UPnP 技术的特点：

1) 支持发现和列举具有网络和服务功能的设备。

2) 支持对等 (peer to peer) 模型结构，在无 PC 的参与下使设备可以直接为其它的设备发现并使用。

3) 扩展了协议功能发现机制，能够列举每一个设备的独特特性，包括通信协议。

4) 建立在一个低成本的微处理器 ASICS 上，只需 1MKBITY 的 RAM 和闪存以及其它很少的系统资源。

5) 基于 XML 的描述原则提供了一种直接、灵活的方式实现设备的功能，不必为新加入的系统资源支付额外的开销。

6) 支持现有的重要工业标准：TCP/IP, HTML, XML, HTTP, DNS, LDAP 等。

UPnP 对两种家庭网络体系的实施方式：

1) 对 AD HOC (Serverless, Peer to Peer) 即对等、无服务器方式的环境中，通过 SLP (Service Location Protocol) 以及 SMB (Server Message Block) 等协议发现并在计算机间共享文件，如果要确定一个 IP 地址是否可得，可以使用 PING 或 ICMP ECHO 等协议，FLOOD-PING 可以用来浏览现在的网络地址以确定那些地址正在使用之中。

2) 对基于服务器配置的系统中采用 DHCP (Dynamic Host Configuraton Protocol) 协议，提供一个将配置信息在 TCP/IP 网络中传递给主机的框架，并分配临时的或长期的 IP 地址给主机，DHCP 是建立在 BOOTP 基础上的，RARP 处理网络中的地址发现问题，并包含一个 IP 地址自动分配机制，系统的配置采用不同的发现服务 (LDAP, NDS)，并使用 SLP 在网络中发现其他的服务。

### 4. 数字家电网络平台内部连网方案

家庭内部网络接口的媒体层方案直接关系到用户的实际使用和安装，也是家庭网络快速发展道路上最迫切需要统一规范的方面，数字家电网络一般可分为三种：高智能信息家电，如电脑，个人数据助理等；视频音频设备，如电视，录像机，电话等；一般电气设备，如空调，电冰箱，微波炉等。该层可供参照的协议有 HomePNA (Home Phoneline Networking Alliance), SWAP (Shared Wireless Access Protocol), IEEE1394, Bluetooth, USB, Powerline 等。

IEEE 1394 由苹果电脑公司提出的 Firewire 标准衍变而来，它是一种灵活易用，低花费，用于信息家电和个人电脑互联的通用串行数字化接口，标准定义了介质拓扑和传输协议，主要侧重于线缆连接应用，是一个硬件、软件集成标准。具有完全的数字接口，无需模拟/数字的转换，减少了数据丢失率；接口较小，串行电缆较细，节省了空间和花费；支持热插拔，用户可以在总线有效状态下插拔 1394 设备；易使用，易维护，安装时无需附加设备和人工端口设置；具有传输速率可升级体系，100Mbit/s、200Mbit/s、400Mbit/s 可在同一总线上依照应用特性随时变更；具有灵活的网络拓扑结构，支持链形、树形的多级桥连；同步通信方式可用于多媒体数据传输，异步通信方式可用于控制信号传输。

Home Phoneline Networking Alliance (HomePNA) 是一种基于电话线的高速、易用、便宜，用于解决家庭中电脑和设备之间通信的网络解决方案，使用简单，维护方便，有较合适

的性价比,支持 Internet 访问,支持接口协议 V.90、ADSL 和 Cable Modem,家庭信息设备通过网络可共享数据和服务,可通过系统的调整和改善使电话线的速率达到 1Mbit/s。Home PNA 的方案尚在制定中,带宽的限制是该项技术发展的不利因素,但由于其网络介质的简易和实用,仍然吸引了大量厂商的参与。

SWAP (Shared Wireless Access Protocol) 是 HomeRF™工作组于 1999 年 1 月 5 日颁布的,SWAP 提供一个开放的家庭信息平台,使各种可互操作的消费电子设备进行无线的音频和数据通信,以建立一个家庭无线网络,主要是利用 PC 在家庭的主导作用来分配和发布 TCP/IP 协议信息解决家庭网络问题。

Bluetooth 是目前发展最为迅速的无线通信协议,它是一种通过无线短程链接传送语音和数据的技术开放性标准,适用于静止或者运动的环境,其目标是用无线替代电话、笔记本电脑等设备的互联缆线,使网络更易于使用,同时开辟例如隐式计算、运动设备的稳定链接等新功能,增加网络条件访问、数据加密等安全措施。技术特点为:工作频率 2.45GHZ,通信半径 10m,可以延展到 100m,传输速率 1Mbit/s,设计适用于多种频率的环境,使用前向纠错 (Forward Error Correction) 限制远程链接的频率干扰。其组成包括语音单元,无线单元链接控制模块,链接管理部件和软件单元。

## 5. 数字家电网络平台的相关标准

### (1) MHP (Multimedia Home Platform) 技术

MHP (多媒体家庭平台) 是 DVB (Digital Video Broadcast) 数字视频广播标准下的一个分协议,DVB 是欧洲广播联盟组织的有 200 多个单位参加的一个项目。它包括卫星,有线电视,地面电视广播,SMATV 和 MVDS 的普通电视和 HDTV 的广播与传输,1997 年 DVB 计划扩展至 MHP。

该方案从服务和应用的角度增强了数据广播,交互式服务和 Internet 访问,其目标是针对目前各公司产品的独立性、不兼容性,试图建立一套基于家庭网络服务的、完整的、平行堆叠的市场层,使各公司的产品在相应层面上有序竞争。

技术特点是:互操作性、向后兼容性、模块化、可扩展性;建立在公开的标准上;支持条件访问;采用通用的 APIS。

其功能是:为现在的条块分割的垂直市场提供途径,使之转化为同一的横向市场;支持由多媒体数据组成的数字电视和传统电视;使用返回通道和交互式服务;支持 Internet 访问;与现有的专用 APIS 并行使用的通用 API;为现有的购物系统提供一个接口,使其与 MHP 并存。

### (2) ATVEF (Advanced TV Enhancement Forum) 交互式电视技术

1998 年 ATVEF 宣告成立,包括 CNN, NBC, PBS, Walt Disney 等大电视节目制作商, Cablelabs, DirectTV, TCI 等主要传送商, Intel, Microsoft, Sony, NCI 等计算机和电子设备商均加入其中。其目标是建立一个受各种电视环境支持的平台,利用已经为 Web 建立起来的各种媒体和工具实现交互式电视的增强广播。

技术特点是:HTML 和相关数据格式用来作为描述图像、文本、电视和其它多媒体的机制;统一资源标识用来作为检索各种多媒体内容的机制;ECMAScript 和文档目标模式用来动态控制 HTML 内容和 HTML 内容对电视广播的同步;IP 用来作为所有数据的基本传输方式,多播 IP 用于单向连接中。

功能特点是：把电视和 Web 页面综合在一起，利用触发器控制 HTML 显示内容的信息，支持单向电视广播网络，定义了两种传送方式，其传送规范可在 Internet 上和任何支持 IP 组播的网络上运行。

### (3) JavaTV

Java TV API 针对数字电视接收器独有的功能而设计，用它编写的交互式电视应用有着极强的安全性、可扩展性和对不同电视接收器的可移植性。该技术包括音频、视频的媒体控制，广播数据的访问，服务信息数据访问，调谐和译码器的控制以及屏幕图形处理，媒体同步和应用软件模块等功能。媒体同步是指电视节目的基本视频和背景音频同步，应用软件模块是指交互式应用软件和传统的电视节目的和谐共存。

JavaTV API 是在现有的标准环境中展开的，这些现有的标准和基于标准的源码集成在 Java TV API 框架中，因此 Java TV API 定义在高于硬件和协议的层次，是 Java 平台的垂直扩展。

Java TV API 运行于数字电视接收器以及相关的设备，数字电视的主要特征是对广播媒体数据通道的控制，这些通道通常拥有通用的功能模块和子系统，比如调谐器和译码器等，Java TV API 应用程序编写人员无需知道有关硬件的细节，就可以编写相应的应用程序。

Java TV API 的定义和相关协议包括关于流选择的广播协议堆，关于流选择的网际互连协议（用户数据协议，传输协议，实时协议），远程通信协议堆，视频解码协议，图形编码协议等。

### (4) OSGI (Open Service Gateway Initiative) 开放服务网关

由 Ericsson, IBM, NCI, Nortel, Oracle, Philips, Sun, Alcatel, Motorola, Lucent, Enron, Cable&Wireless, EDF, Sybase 和 Toshiba 建立。

其功能是：作为各种基于通信的服务平台；将提供和管理家庭和小型办公室收发的语音、数据、互联网和多媒体通信；成为大量高值服务的应用服务器，如能源管理和控制、保安服务、健康监测服务、设备控制和维护、电子商业服务等等。

它的技术特点：保证全新种类的设备如 Jini 和 HAVi 的互连和管理，也集成了全部或部分已存在的产品，如（数字或模拟）机顶盒，线缆调制解调器，集线器，家用网关，报警系统，能源管理系统，消费电子，个人计算机等。

这种服务网关（SG）将采用已有的 Java 标准如 Jini，并集成其他非 Java 标准如 HAVi。把这些设备标准连接到中心管理系统，并为服务提供商提供网关，方便服务的设置。

这种服务网关是一种集成服务器，内置网络功能连接外部互联网和内部客户。

这种服务网关插入在服务提供商网络和家庭，或 SOHO/ROBO 局域网和客户设备之间 SG 处于外部网络和内部网络之间。

服务由外部互联网上可信赖的服务提供商发到 SG 或内部客户。

SG 是典型的零管理设备，安全工作在内外器件之间。

OSGI 参数包括服务全周期管理，内部服务基础，数据管理，设备管理，客户操作，资源管理和安全性的 APIs。使用这些 APIs 的客户在需要时从服务提供商载入网络的服务，并由 SG 管理这些服务的安装，版本和配置。

数字家电网络平台技术是当今计算机技术从体系结构、操作系统到驱动模式、网络应用



模式最新成果的集成体，倍受 IT 业界的青睐，但在其迅猛发展的同时，也应当看到它作为新兴的应用技术，不可避免地存在产品结构的差异性、功能的不完备性以及性能不确定性等问题，也正是这些亟待解决的问题，又反过来推动计算机技术的进一步发展，从而为整个信息产业的发展带来了新的契机。

## 1.3 多媒体计算机技术的发展和應用

1994 年，我国国家经贸委经过充分论证，将多媒体技术列入国家级技术开发重点项目计划，并给予了高度重视，尤其在多媒体基础技术、多媒体平台及多媒体应用等方面给予了重点资金支持。因此，我国在多媒体芯片和板级产品、CD 系列数字影碟机、多媒体汉语语音交互技术、DVD 高密度数字光盘及多媒体通信计算机等方面有了长足的进步，并涌现了一批在多媒体领域效益较高的企业，成为计算机产业新的增长点之一。

目前国内外多媒体计算机研究人员及有关厂商，正在多媒体数据库、多媒体通信及多媒体应用等方面积极开展研究工作，并有部分产品推向市场。

### 1.3.1 多媒体数据库

传统的数据库管理系统在处理结构化的数据如文字、数值等信息方面取得了很大成功，然而在很多应用领域，如 CAI 课件、办公自动化、诊断医疗管理系统、图书馆和博物馆管理系统、计算机辅助设计及地理信息系统等，由于这些应用包含了多种媒体数据和非结构化数据，传统的数据库管理系统就显得有些力不从心。

#### 1. 多媒体数据库的研究途径

目前多媒体数据库的研究主要有以下三条途径：

- 在现有商用数据库管理系统的基础上增加接口，以满足多媒体应用的需要。
- 建立基于一种或几种应用的专用多媒体信息管理系统。
- 从数据模型入手，研究全新的通用多媒体数据库管理系统。

第一种途径实用，但是效率很低；第二种途径易于实现，但缺乏通用性，而且可扩展性差；第三种途径是研究和发展的主流，但是具有相当的难度。

#### 2. 多媒体数据库要解决的关键技术问题

##### (1) 数据模型

多媒体数据模型主要采用关系数据模型的扩充和采用面向对象的设计方法。由于用传统的关系模型难以描述多媒体信息和定义对多媒体数据对象的操作，目前在关系模型扩充方面除了引入抽象数据类型外，较多的采用语义模型的方法。关系模型主要描述数据的结构，而语义模型则主要表达数据的语义，语义模型的层次高于关系模型，后者可以作为前者的基础。目前的研究表明，采用面向对象的方法来描述和建立多媒体数据模型是较好的方法，面向对象的主要概念包括对象、类、方法、消息、封装和继承等，可以方便地描述复杂的多媒体信息。

##### (2) 数据的压缩和解压缩

由于多媒体数据，如声音、图像及视频等数据量大，存储和传输需要很大的空间和时间，因此必须考虑对数据进行压缩编码，压缩方法要考虑到复杂性，实现速度及压缩质量等问题。

##### (3) 多媒体数据的存储管理和存取方法

目前常用的有分页管理、B\*树和 Hash 方法等。在多媒体数据库中还要引入基于内容的检索方法、矢量空间模型信息索引检索技术、超位检索技术及智能索引技术等。

#### (4) 多媒体信息的再现及良好的用户界面

在多媒体数据库中应提供多媒体宿主语言调用，还应提供对声音、图像、图形和动态视频的各种编辑和变换功能。

#### (5) 分布式技术

多媒体数据通信对网络带宽有较高的要求，需要相应的高速网络，此外还要解决数据集成、异构多媒体数据语言查询、调度和共享等问题。

由于研究多媒体数据库在理论上和实践上都存在较大困难，因此国内外目前研制开发许多商品化的系统都只能称作为多媒体信息管理系统，因为它们具备了管理多种媒体的能力，但离理想的多媒体数据库还有一定差距。多媒体数据库不只是具有存储管理多媒体信息的能力，而应该能把多种媒体统一起来，支持对各种媒体信息的语义查询和检索。

### 1.3.2 多媒体通信

现代化社会人的工作方式的特点是具有群体性、交互性、分布性及协作性。传统的电信业务电话，传真等通信方式已不能适应社会的需要，为了提供更具有人性化的交流环境，把通信手段从语音为主转向视频为主是一个很自然的要求。笔者认为多媒体通信可以分成两类：一类是对称的全双工的多媒体通信，如分布式多媒体信息系统、视频会议系统及计算机支持的协同工作系统；另一类是非对称全双工的多媒体通信系统，如交互式电视系统（ITV）、点播电视系统（VOD）、远程教育系统、远程医疗诊断系统及远程图书馆。

对于多媒体通信要解决两个关键技术：多媒体数据压缩和高速数据通信问题。尤其是实用化效果较好，应用比较广泛的视频会议系统要解决视频会议系统的国际标准问题。标准化是产业化成功的前提，这样用户可以把不同厂家的不同产品连接在一起，实现彼此间的相互通信。

ITU-T（国际通信联盟标准化委员会）制定的 H.320 协议标准是 ISDN 视频会议市场占主导地位的标准，现在 ITU-T 委员会正在把这套标准扩大到包括多点呼叫标准（T.120）、计算机图形标准以及模拟电路视频会议适用的低速率电路标准（H.324）。Intel 公司与 150 个计算机和通信公司成立了个人会议工作小组（PCWG），并于 1994 年制定了个人会议标准（PCS-Personal Conferencing Specification），其目的是保证基于文本的会议可以在各种操作系统、硬件平台和传输媒体中互操作。视频会议系统可分为两类：点对点视频会议系统和多点视频会议系统。

点对点视频会议系统有可视电话、台式机—台式机视频会议及会议室—会议室视频会议。多点视频会议系统允许三个或三个以下不同地点的参加者一起参加讨论。它的关键技术是多点控制器（MCU），它能自动地交换数据，把正确的音频和视频信号发送给每个与会者，多点控制器可以被放置在视频会议网络的任何一个点上，它通过编码和解码器接收所有的数字信号，并自动地把数据发送到合适的地点。

### 1.3.3 多媒体创作工具及其应用

多媒体创作工具是电子出版物、多媒体应用系统的开发工具，它提供组织和编辑电子出

版物 and 多媒体应用系统各种成分所需要的重要框架,包括图形、动画、声音和视频的剪辑。制作工具的用途是建立具有交互式的用户界面,在屏幕上演示电子出版物及制作好的多媒体应用系统以及将各种多媒体成分集成为一个完整而有内在联系的系统。

多媒体创作工具可以分成基于时间的创作工具;基于图符 (Icon) 或流线 (Line) 的创作工具;基于卡片 (Card) 和页面 (Page) 的创作工具;以传统程序语言为基础的创作工具。它们的代表产品是 Action、Authorware、Icon Author、Tool Book 以及 HyperCard。

我们希望多媒体创作工具具有下述功能:具有良好的、面向对象的编程环境;具有较强的支持多媒体数据 I/O 能力;能播放由外部程序制作好的动画并具有简单的动画制作和处理功能;具有超级连接能力 (Hypertext);具有应用程序能够调用另一个处理函数的能力;具有形成安装文件或可执行文件的功能以及易学易用并有良好的技术支持。

用多媒体创作工具可以制作各种电子出版物及各种教材、参考书、导游和地图、医药卫生、商业手册及游戏娱乐节目;多媒体应用系统;演示系统或信息查询系统;培训和教育系统;娱乐、视频动画及广告;专用多媒体应用系统等。目前在国内市场上销售的多媒体应用系统有:合力电子技术公司研制的“房地产售楼咨询系统”;北京海达国际电脑软件工程公司推出的“768 多媒体电视声文图节目自动制造及播放系统”及“触摸屏声文图信息系统”;大恒图像视觉有限公司研制的“多媒体广告制作及演示系统”、“太极多媒体触摸查询系统”,可用于大型运动会、实时信息查询、领导决策辅助系统、饭店信息查询系统、导游系统、歌舞厅点歌结算系统、商店导购系统、生产商业实时监测系统以及证券交易实时查询系统。

### 1.3.4 多媒体计算机的发展趋势

#### 1. 进一步完善计算机支持的协同工作环境 (CSCW, Computer Supported Collaborative Work)

目前多媒体计算机硬件体系结构,多媒体计算机的视频音频接口软件不断改进,尤其是采用了硬件体系结构设计和软件、算法相结合的方案,使多媒体计算机的性能指标进一步提高。但要满足计算机支撑的协同工作环境的要求,还需进一步研究。多媒体信息空间的组合方法,要解决多媒体信息交换,信息格式的转换以及组合策略。由于网络延迟,存储器的存储等待,传输中的不同步以及多媒体等时性的要求等,因此还需要解决多媒体信息的时空组合问题,系统对时间同步的描述方法以及在动态环境下实现同步的策略和方案。这些问题解决后,多媒体计算机将形成更完善的计算机支撑的协同工作环境,消除了空间距离的障碍,也消除了时间距离的障碍(可以充分享用历史的设计资料),为人类提供更完善的信息服务。

#### 2. 智能多媒体技术

1993 年 12 月,英国计算机学会在英国 Leeds 大学举行了多媒体系统和应用 (Multimedia System and Application) 国际会议。英国 Michael D. Vislon (Rutherford Appleton laboratory) 在会上作了关于建立智能多媒体系统的报告,明确提出了研究智能多媒体技术问题。作者认为,多媒体计算机充分利用了计算机的快速运算能力,综合处理声、文、图信息,用交互式弥补计算机智能的不足。进一步的发展就应该是增加计算机的智能,根据中国的国情和现状,切实可行的方案是使多媒体计算机增加如下的智能:

(1) 文字的识别和输入。印刷体汉字、联机手写体汉字以及脱机手写体汉字的识别和

输入。

(2) 汉语语音的识别和输入。主要是特定人、非特定人以及连续汉语语音的识别和输入。

(3) 自然语言理解和机器翻译。汉语的自然语言理解和机器翻译。

(4) 图形的识别和理解。

(5) 机器人视觉和计算机视觉。

(6) 知识工程以及人工智能的一些课题。

目前,国内有的单位已经初步研制成功了智能多媒体数据库,它的核心技术是将具有推理功能的知识库与多媒体数据库结合起来形成智能多媒体数据库。另一个重要的研究课题是多媒体数据库基于内容检索技术,它需要把人工智能领域中的高维空间的搜索技术、视音频信息的特征抽取和识别技术、视音频信息的语义抽取问题以及知识工程中的学习、挖掘及推理等问题应用到基于内容检索技术。

总之,把人工智能领域某些研究课题和多媒体计算机技术很好地结合,就是多媒体计算机长远的发展方向。

### 3. 把多媒体信息实时处理和压缩编码算法做到 CPU 芯片中

计算机产业的发展趋势应该是把多媒体和通信的功能集成到 CPU 芯片中,过去计算机结构设计较多地考虑计算功能,主要用于数学运算及数值处理,最近几年随着多媒体技术和网络通信技术的发展,需要计算机具有综合处理声、文、图信息及通信的功能。经过大量的实验分析多媒体信息的实时处理、压缩编码算法及通信,大量运行的是 8 位和 16 位定点矩阵运算。把这些功能和算法集成到 CPU 芯片中要遵循下述几条原则:

(1) 压缩的算法采用国际标准的设计原则。

(2) 多媒体功能的单独解决变成集中解决。

(3) 体系结构设计和算法相结合。

为了使计算机能够实时处理多媒体信息,对多媒体数据进行压缩编码和解码,最早的解决办法是采用专用芯片,设计制造专用的接口卡。最佳的方案应该把上述功能集成到 CPU 芯片中。从目前的发展趋势看可以把这种芯片分成两类:一类是以多媒体和通信功能为主,融合 CPU 芯片原有的计算功能,它的设计目标是用在多媒体专用设备、家电及宽带通信设备,以取代这些设备中的 CPU 及大量 ASIC 和其他芯片。它们的代表产品是 Philips 公司的 Trimedia, MicroUnity 的 Media Processor 及 Chromatic Research Inc. 的 Mpaact Media Engine。另一类是以通用 CPU 计算功能为主,融合多媒体和通信功能,它们的设计目标与现有的计算机系列兼容,同时具有多媒体和通信功能,主要用在多媒体计算机中。它们的代表产品是 Sun 公司的 Ultra Sparc I 和 II, Cyrix Multimedia 586, HP 公司的 MAX-2, Intel 公司的 MMX 及 Motorola 公司的 Ve Comp701 等。

## 1.4 小结

本章对多媒体计算机的定义、分类和多媒体计算机要解决的关键技术以及多媒体技术的应用和发展,均作了详细的讨论。

多媒体计算机技术综合处理声、文、图、音频、视频等信息。多媒体计算机具有信息载

体的多样性、集成性和交互性。

多媒体技术促进了通信、娱乐和计算机的融合，为解决电视数字化高清晰度电视提供了切实可行的方案。应用多媒体计算机技术可制作 VCD、DVD 及影视音响设备以及数字家电网络平台。多媒体技术的发展促进了多媒体数据库、多媒体通信、多媒体创作工作及应用。多媒体计算机将朝着高分辨率提高显示质量、高速化、简单化、智能化方向发展。

## 1.5 习题

1. 请根据多媒体的特性判断以下哪些属于多媒体的范畴？

- (1) 交互式视频游戏
  - (2) 有声图书
  - (3) 彩色画报
  - (4) 立体声音乐
- (A) (1) (B) (1)、(2)  
(C) (1)、(2)、(3) (D) 全部

2. 下列哪些不是多媒体核心软件？

- (1) AVSS
  - (2) AVK
  - (3) DOS
  - (4) Amiga Vision
- (A) (3) (B) (4)  
(C) (3) (4) (D) (1) (3)

3. 要把一台普通的计算机变成多媒体计算机，要解决的关键技术是什么？

- (1) 视频音频信号的获取
  - (2) 多媒体数据压缩编码和解码技术
  - (3) 视频音频数据的实时处理和特技
  - (4) 视频音频数据的输出技术
- (A) (1) (2) (3) (B) (1) (2) (4)  
(C) (1) (3) (4) (D) 全部

4. Commodore 公司在 1985 年率先推出了第一个多媒体计算机 Amiga，其主要功能是：

- (1) 用硬件显示移动数据，允许高速的动画制作；
  - (2) 显示同步协处理器；
  - (3) 控制 25 个通道的 DMA，使 CPU 以最小的开销处理盘、声音和视频信息；
  - (4) 从 28Hz 振荡器产生系统时钟；
  - (5) 为视频 RAM (VRAM) 和扩展 RAM 卡提供所有的控制信号；
  - (6) 为 VRAM 和扩展 RAM 提供地址。
- (A) (1)、(2)、(3) (B) (2)、(3)、(5)  
(C) (4)、(5)、(6) (D) 全部

5. 国际标准 MPEG—II 采用了分层的编码体系，提供了四种技术，它们是：

- (1) 空间可扩展性；信噪比可扩充性；框架技术；等级技术。
- (2) 时间可扩充性；空间可扩展性；硬件扩展技术；软件扩展技术。
- (3) 数据分块技术；空间可扩展性；信噪比可扩充性；框架技术。
- (4) 空间可扩展性；时间可扩充性；信噪比可扩充性；数据分块技术。

(A) (1)

(B) (2)

(C) (3)

(D) (4)

6. 多媒体技术未来发展的方向是什么？

(1) 高分辨率，提高显示质量。

(2) 高速度化，缩短处理时间。

(3) 简单化，便于操作。

(4) 智能化，提高信息识别能力。

(A) (1)、(2)、(3)

(B) (1)、(2)、(4)

(C) (1)、(3)、(4)

(D) 全部

7. 数字家电网络平台现有的相关标准有那些？

8. 简述多媒体计算机的关键技术及其主要应用领域。

## 1.6 参考答案

1. (B)

2. (A)

3. (D)

4. (D)

5. (D)

6. (D)

7. (略)

8. (略)

## 第2章 声卡和音频信息

### 本章要点

- 数字化音频的获取与处理基本概念，模拟音频与数字音频的区别。数字音频采样和量化的基本原理以及数字音频的文件格式和音频信号的特点。
- 音频卡的工作原理以及音频卡的功能和分类，音频卡的安装使用。
- 音频编码的原理和标准以及编码解码的基本方法。

声音是携带信息的极其重要的媒体，是多媒体技术研究中的一个重要内容。声音的种类很多，可分为语音（言语）和非语音（乐音和杂音）。声音是人们用来传递信息最方便、最熟悉的方式，声音携带的信息量大、精确、准确。

随着多媒体信息处理技术的发展，计算机数据处理能力的增强，音频处理技术受到重视，并得到了广泛的应用。如视频图像的配音、配乐；静态图像的解说、背景音乐；可视电话、电视会议中的话音；游戏中的音响效果；虚拟现实中的声音模拟；用声音控制 Web，电子读物的有声输出等。

### 2.1 数字音频基础

#### 2.1.1 模拟音频和数字音频

众所周知，模拟磁性录音技术已经使用多年。这一技术的原理广泛地用于采集、播放各种各样的声音，如音乐、配音及特殊的声效果。这种模拟录音方式是直接记录音频信号的波形，重放时用唱针扫描槽纹或者用放音磁头来拾取信号。但模拟磁性录音性能受电磁性能的影响较大，磁带的频率特性微小的变化都会对音质产生影响。目前模拟录音的动态范围可达 80dB。若想进一步提高录音、放音的音质，只能求助于数字音频技术。声音是机械振动，振动越强，声音越大。话筒把机械振动转换成电信号。模拟音频技术以模拟电压的幅度来表示声音强弱。

数字音频系统是通过将声波波形转换成一连串的二进制数据来再现原始声音的，实现这个步骤使用的设备是模/数转换器（A/D）。它以每秒上万次的速率对声波进行采样，每一次采样都记录下了原始模拟声波在某一时刻的状态，称之为样本。

将一串样本连接起来，就可以描述一段声波了，每一秒钟所采样的数目称为采样频率或采率，单位为 Hz（赫兹）。采样频率越高所能描述的声波频率就越高。对于每个采样，系统均会分配一定存储位（bit 数）来表达声波的振幅状态，称之为采样分辨率或采样精度，每增加一个 bit，表达声波振幅的状态数就翻一翻。可以计算出 16bit 能够表达 65536 种状态。采样精度越高，声波的还原就越细腻。

在数字音频技术中，把表示声音强弱的模拟电压用数字表示。如 0.5V 电压用数字 20 表示，2V 电压用 80 表示。模拟电压幅度，即使在某电平范围内，仍然可以有无穷多个，如 1.2V，1.21V，1.215V…。而用数字来表示音频幅度时，只能把无穷多个电压幅度用有限个数字表示。即把某一幅度范围内的电压用一个数字表示，这称作量化。

模拟声音在时间上是连续的，而以数字表示的声音则是一个数据序列，在时间上只能是断续的。因此当把模拟声音变成数字声音时，需要每隔一个时间间隔在模拟声音波形上取一个幅度值，这称为抽样。时间间隔称为抽样周期（其倒数称为采样频率）。

由此看出，数字声音是一个数据序列。它是模拟声音经抽样、量化后得到的。计算机、数字 CD、数字磁带（DAT）中存储的都是数字声音。模拟—数字转换器可以把模拟声音变成数字声音；数字—模拟转换器可以恢复出模拟声音。

## 2.1.2 音频的数字化

目前，多媒体计算机产生声音的方式主要有三种：由外部声音源进行录制与重放；MIDI 音乐和 CD-Audio，用 Windows 的术语，分别称为波形音频、MIDI 音频和 CD 音频。

数字化主要包括采样和量化两个方面；相应地，数字化音频的质量取决于采样频率和量化位数这两个重要参数。

声音数字化的两个步骤是采样和量化。采样就是每间隔一段时间就读一次声音信号的幅度，量化就是把采样得到的声音信号幅度转换为数字值。时间上的离散叫采样，幅度上的离散称为量化。

### 1. 采样频率（Sampling Rate）

采样频率是指将模拟声音波形数字化时，每秒钟所抽取声波幅度样本的次数，采样频率的计算单位是 kHz（千赫兹）。一般来讲，采样频率越高声音失真越小，但用于存储音频的数据量也越大。

采样频率的高低是根据奈奎斯特采样定理（Nyquist theory）和声音信号本身的最高频率决定的。奈奎斯特采样定理：设连续信号  $x(t)$  的频谱为  $x(f)$ ，以采样间隔  $T$  采样得到离散信号  $x(nT)$ ，如果满足：当  $|f| \geq f_c$  时， $f_c$  是截止频率  $T \leq 1/2f_c$  或  $f_c \leq 1/2T$  则可以由离散信号  $x(nT)$  完全确定连续信号  $x(t)$ 。当采样频率等于  $1/(2T)$  时，即  $f_N = 1/2T$ ，称  $f_N$  为奈奎斯特频率。奈奎斯特采样定理指出，采样频率不应低于声音信号最高频率的两倍，这样就能把以数字表达的声音还原为原来的声音。

正常人耳听觉的频率范围大约在 20Hz~20kHz 之间，根据采样理论，为了保证声音不失真，采样频率应在 40kHz 左右。常用的音频采样率有：8kHz、11.025kHz、22.05kHz、16kHz、37.8kHz、44.1kHz、48kHz 等。如果采用更高的采样频率，还可以做出 DVD 的音质。

### 2. 量化数据位数（也称量化级，样本尺寸等）

量化位数是每个采样点能够表示的数据范围，常用的有 8 位、12 位和 16 位。例如，8 位量化级表示每个采样点可以表示 256 个（0~255）不同量化值，而 16 位量化则可表示 65536 个不同量化值。量化级的大小决定了声音的动态范围，即被记录和重放的声音最高与最低之间的差值。16 位的量化级足以表示从人耳刚刚听得见的极细微的声到感觉难以忍受的巨大噪声这样大的声音范围。同样，量化位数越高音质越好，数据量也越大。图 2-1 是对音频数字化过程的简要说明。



量化的过程是，先将整个幅度划分成为有限个小幅度（量化阶距）的集合，把落入某个阶距内的样值归为一类，并赋予相同的量化值。

量化有很多方法，但可以归纳为两类：一类称为均匀量化，另一类称为非均匀量化。采用的量化方法不同，量化后的数据量也不同。因此，可以说量化也是一种压缩数据的方法。

### （1）均匀量化

采用相等的量化间隔对采样得到的信号做量化就是均匀量化。均匀量化就是采用相同的“等分尺”来度量采样得到的幅度，也称为线性量化。

用这种方法量化输入信号时，无论对大的输入信号还是对小的输入信号一律都采用相同的量化间隔。为了适应幅度大的输入信号，同时又要满足精度高的要求，就需要增加样本的位数。但是，对语音信号来说，大信号出现的机会并不多，增加的样本位数就没有充分利用。为了克服这个不足，就出现了非均匀量化的方法，这种方法也称为非线性量化。

### （2）非均匀量化

非均匀量化的基本思想是，对输入信号进行量化时，大的输入信号采用大的量化间隔，小的输入信号采用小的量化间隔，这样就可以在满足精度要求的情况下使用较少的位数来表示。声音数据还原时，采用相同的规则。

在非线性量化中，采样输入信号幅度和量化输出数据之间定义了两种对应关系，一种称为 $\mu$ 律压扩算法，另一种称为A律压扩算法。

## 3. 单声道与双声道

反映音频数字化质量的另一个因素是通道（或声道）个数。记录声音时，如果每次生成一个声波数据，称为单声道的；每次生成二个声波数据，称为立体声（双声道），立体声更能反映人的听觉感受。除了上述因素外，数字化音频还受其他一些因素（如扬声器的质量）的影响。

## 4. 数字音频的存储

可以用下面的公式估算声音数字化后每秒所需的存储量（假定不经压缩）：

wav 文件每秒的存储量（字节）= 采样频率（Hz）× 量化位数（位）× 声道数/8

例如，数字激光唱盘（CD-DA，红皮书标准）的标准采样频率为 44.1kHz，量化位数为 16 位，立体声（这就是所谓的 CD 音质——CD-Quality Sound），可以几乎无失真地播出频率高达 22kHz 的声音，这也是人耳所能听到的最高声音频率。一分钟 CD-DA 音乐所需的存储量为：

$(44.1 \times 1000 \times 16 \times 2 \times 60/8)$  字节 = 10 584 000 字节

## 2.1.3 数字音频的文件格式

在多媒体计算机中，存储声音信息的文件格式主要有：WAV 文件、VOC 文件、MIDI 文件、AIF 文件、SNO 文件及 RMI 文件等。

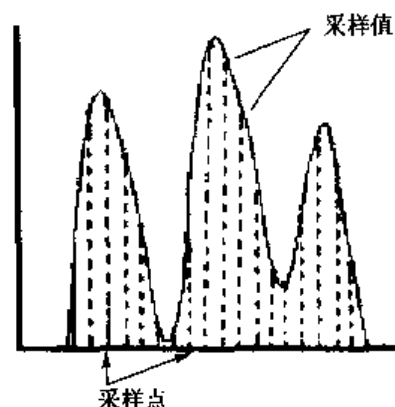


图 2-1 声音波形的采样和量化

## 1. 波形音频

波形音频是多媒体计算机获得声音最直接、最简便的方式。在这种方式中，通常以麦克风、立体声录音机或CD激光唱盘等作为声音信号的输入源，声卡以一定的采样频率和量化级对输入声音进行数字化，将其从模拟声音信号转换为数字信号（模/数转换），然后以适当的格式存在硬盘上。记录下来的声音重放时，声卡将文件中的数字信号还原成模拟信号（数/模转换），经混音器混合后由扬声器输出。

波形文件是 Windows 所使用的标准数字音频文件，文件的扩展名是 WAV，记录了对实际声音进行采样的数据。在适当的硬件及计算机控制下，使用波形文件能够重现各种声音，包括不规则的噪声、CD 音质的音乐、单声道或立体声。多数声卡都能以 16 位量化级、44.1kHz 的采样率（CD 音质）录制和重放立体声声音。

波形文件的主要缺点是产生的文件太大，不适合长时间记录。wav 文件是由采样数据组成的，所以它需要的存储容量很大。当然，如果对声音质量要求不高，则可以通过降低采样频率，采用较低的量化位数或利用单音来录制 wav 文件，此时的 wav 文件大小可以成倍地减小。

由于原始声音数据量太大，有必要采用硬件或软件方法进行压缩处理，常用的软件压缩方法主要有 ACM（Microsoft's Audio Compression Manager）和 PCM 等。另一方面，一般人的讲话声使用 8 位量化级，11.025kHz 采样率就能较好地还原（如电话声），因此，这种质量较低的波形文件在应用中也不少见。

通过 Windows 的对象连接与嵌入（OLE）技术，波形文件可以嵌在其他 Windows 应用系统中使用。由于波形文件记录的是数字化音频信号，因此可由计算机对其进行处理和分析，如放慢或加快放音速度、将声音重新组合或抽取出一一些片断单独处理等等，Windows 中的“录音机”（Sound Recorder）就是一个方便的工具。

## 2. VOC 文件

VOC 文件是 Creative 公司波形音频文件格式，也是声霸卡（sound blaster）使用的音频文件格式。每个 VOC 文件由文件头块（header block）和音频数据块（data block）组成。文件头包含一个标识、版本号和一个指向数据块起始的指针。数据块分成各种类型的子块，如声音数据、静音、标记、ASCII 码文件，重复以及终止标志、扩展块等。

VOC 格式音频文件的文件头如下：

- 00H~13H 字节。文件类型说明。前 19 个字节包含正文：Creative Voice File。最后是 EOF 字节（1AH）。
- 14H~15H 字节。其值为 001AH。
- 16H~17H 字节。文件的版本号。小数点后面的部分在前。如版本号为 1.10，则这两个字节内的值为 0A01。
- 18H~19H 字节。是一个识别码。由这个代码可以检验其文件是否是真正的 VOC 文件。其值是 16H 和 17H 单元中所存文件版本号的反码再加上 1234H。例如，版本号为 1.10，010AH 的反码是 FEF5H，则这个代码为：FEF5H+1234H=1129H。

利用声卡提供的软件可以实现 VOC 和 WAV 文件的转换。

## 3. MIDI 文件

MIDI 音频是多媒体计算机产生声音（特别是音乐）的另一种方式，可以满足长时间音

乐的需要。

由于 MIDI 文件记录的并不是声音本身,因此它比较节省空间。与波形文件不同的是, MIDI 文件(扩展名为.MID)并不对音乐进行采样,而是将每个音符记录为一个数字, MIDI 标准规定了各种音调的混合及发音,通过输出装置就可以将这些数字重新合成为音乐。与波形文件相比, MIDI 文件要小得多;例如,同样半小时的立体声音乐, MIDI 文件只有 200KB 左右,而波形文件(.WAV)则要差不多 300MB。

MIDI 格式的主要限制是它缺乏重现真实自然声音的能力,因此不能用在需要语音的场合(这时要与波形文件合用)。此外, MIDI 只能记录标准所规定的有限种乐器的组合,而且回放质量受声卡上合成芯片的严重限制,难以产生真实的音乐演奏效果。近年国外流行的声卡普遍采用波表法进行音乐合成,使 MIDI 音乐的质量大大提高(效果接近 CD 音质),但波表卡仍较昂贵,在我国还未普及。

#### 4. CMF 文件

CMF 文件(Creative Music File)也是随声卡一起诞生的,是它自带的 MIDI 文件存储格式。

#### 5. CD 音频

符合 MPC2 标准的 CD-ROM 驱动器不仅可以读取 CD-ROM 盘的信息,还能播放数字 CD 唱盘(CD-DA),这样多媒体计算机就能够利用已经非常成熟的数字音响技术来获得高质量的音频——CD 音频。CD 音频也是一种数字化声音,以 16 位量化级、44.1kHz 采样率的立体声存储,可完全重现原始声音,每片 CD 唱盘能记录约 74min 这种质量的音乐节目。

在多媒体计算机上输出 CD 音频信号一般有两种途径,一种是通过 CD-ROM 驱动器前端的耳机插孔输出,另一种使用特殊连线接入声卡放大后由扬声器输出。前者的输出音质不受声卡质量的影响,但不能使用声卡的混声功能;而后者虽然可以与波形或 MIDI 音频进行混音输出,但声卡的放大功率都比较小(一般不超过 4W),通常需要有源扬声器或配置外部声音放大器来获得足够的音量。

### 2.1.4 音频信号的特点

在多媒体系统中,音频信号可分为两类:语音信号和非语音信号。非语音信号又可分为乐音和杂音。非语音信号的特点是不具有复杂的语意和语法信息,信息量低、识别简单。语音是语言的物质外壳(载体)。语言是人类社会特有的一种信息系统,是社会交际工具的符号。

音频信号处理的特点如下:

(1) 音频信号是时间依赖的连续媒体。因此音频处理的时序性要求很高。如果在时间上有 25ms 的延迟,就会感到断续。

(2) 由于人接收声音有两个通道(左耳、右耳),因此为使计算机模拟自然声音,也应有两个声道,即理想的合成声音应是立体声。

(3) 由于语音信号不仅仅是声音的载体,同时还携带了情感的意向,故对语音信号的处理不仅是信号处理问题,还要抽取语意等其他信息,因此可能会涉及到语言学、社会学、声学等。

从人与计算机交互的角度来看音频信号相应的处理如下:

(1) 人与计算机通信(计算机接收音频信号):

音频获取: 语音识别与理解。

(2) 计算机与人通信(计算机输出音频):

音频合成: 包括音乐合成和语音合成。

声音定位: 包括立体声模拟、音频/视频同步, 目的是让计算机产生真实感声音。

(3) 人通过计算机与别人通信:

人通过网络与处于异地的人进行语音通信, 需要的音频处理包括: 语音采集、音频编码/解码、音频传输等。这里音频编/解码技术是信道利用率的关键。

## 2.1.5 3D 音频

在一向讲究软硬兼施的 PC 界, 多声道音频的实现自然少不了软件算法的控制过程, 因此, 3D 音频 API 就扮演了重要的接口角色。当然, 有些 API 还包含了具体算法。这些 API 与 3D 图形程序接口, 统称为 3D API (Application Program Interface), 即 3D 应用程序接口。对于支持 3D 定位技术的新一代声卡而言, 算法往往决定了其定位及其它效果的优劣, 因此我们有必要对它们的编程接口有一些简单的了解。

音频 API 种类繁多, 目前各种游戏可以使用的 API 和 3D 技术大体上有 A3D、DirectSound 3D (DS3D)、EAX、Sensaura 3D、Q3D、9-1AS 等等。不同的声卡硬件和不同的游戏往往支持多种不同的 API 和 3D 技术, 这主要取决于声卡所采用的音效芯片的类型。

### 1. DirectSound 3D

它是 DirectX 中的一个组件, 是 Microsoft 公司专为游戏开发的 API, 得益于 DirectX 的不断发展和完善, DS3D 得到了众多声卡厂商的支持。DS3D 的作用在于帮助开发者定义声音在 3D 空间中的定位和声响, 然后把它交给与 DS3D 兼容的声卡, 让他们用各种算法去实现。定位声音的效果好坏实际上取决于声卡所采用的算法。

### 2. Aureal 3D

简称 A3D, 是由 Aureal Semiconductor 公司开发的新型 3D 音效定位技术, 使用这一技术的应用程序(通常是游戏)可以根据用户的选择而决定音效的变化, 而且可以只用一对普通的音箱或耳机来实现, 产生围绕听者的 3D 精确定位音效。

Aureal 推出的最新一代 3D 定位音效标准 A3D 2.0, 还支持对音效 48kHz 频谱的 3D 处理, 另外一项关键的进步是采用了实时声学反射、回音和阻塞渲染技术在内的声波追踪 (Wavetracing) 技术。声学环境的几何描述和墙面的材质特性都可在新的 A3D2.0 的 API 中反映出来。

### 3. EAX

就是环境音效扩展集 (Environmental Audio Extensions)。它的本质是一种依赖于 Microsoft 的 DirectSound3D 的开放 API, 任何人都可以使用这一接口来开发或者在自己的软硬产品中加入对 EAX 的支持。在最新发布的 EAX3.0 中, 加入了功能强大、简单易用的可以为每一个单独音源做反射和混响控制、局限反射群等特效设计工具, 并为开发者公开了全部的环境音效参数, 这对电脑音乐迷具有很大的帮助。

### 4. Sensaura

支持 DS3D, 并且在它们的 DS3D 驱动程序中包含了一个 Voice Manager。开发者可以

用来选择最重要的音源使用 3D 模式，而其余的使用立体声模式。Sensaura 也支持 EAX，并已为一些声卡（如 Yamaha 的 WaveForce）发布了 EAX 驱动程序。

Sensaura 还为解决当前的 HRTE（Head-Related-Transfer-Function）不能很好解决听者在 1m 范围内定位声音的问题而开发了名为 MacroFX 的新技术。

## 5. Qsound

和 Sensaura 一样，只提供音效技术，它推出的 Q3D 技术同样可以用两个喇叭或耳机产生 3D 音效。使用 Q3D 技术的声卡支持 DS3D、EAX 和 A3D 1.X（像 Creative 和 Sensaura 一样，A3D 的调用被转化为 DS3D 调用）。Q3D 不仅使用于游戏，事实上，Qsound 用 Q3D 技术产生了一种杜比认证的虚拟多通道技术——Qsurround，这项技术在家电产品上得到使用。

## 6. IAS

上面这么多的 API 和技术各有特点，这就必须针对不同的系统和 API 编写多套代码，IAS（Interactive Around-Sound）就是针对这个特点而形成的。

IAS 是 Extreme Audio Reality Inc 公司开发的专利音频技术，这个技术能测试系统硬件，管理所有的音效平台需要，因而开发者可以只写一套音效代码，所有基于 Windows 的音频硬件将通过同样的编程界面来获得支持。IAS 提供了 DS3D 支持和其它环绕声的执行程序。

# 2.2 声卡的组成与工作原理

在还没有发明声卡的时候，PC 游戏是没有任何声音效果的。即使有，那也是从 PC 机的一个简单的扬声器里发出的那种“滴里搭拉”的刺耳声，它是用来提醒人们注意的。虽然效果差，但在那个时代这已经令人非常满意了。为了得到更好的声音效果，人们进行了大量的研究和实验，最后终于诞生了声卡。

真正意义上的第一块声卡是由 Adlib Audio 公司于 1984 年研发的，就连现在的 Windows 视窗中还有为这块声卡保留的驱动程序，它的地位如显卡领域中的“Voodoo”一样显赫，即使现在已没有人使用它。

电脑的第一次发声不是在现在大红大紫的 PC 上，而是在 APPLE 的机种上。当时苹果公司的工作人员在一次记者招待会上为大家演示了一段由电脑发出的语音，虽然那样的效果实在不敢令人恭维，但在那个“哑巴计算机的年代”，仍然相当地吸引人。

如今，有上百万的 PC 配备了音频能力，一些系统出售时本身就带有音频功能，如 Compaq 的 Deskpro/I 和一些其他销售商的 MPC 机。另外一些系统备有可扩展的音频端口（可连到 PC 的并行口上）或音频卡。

## 2.2.1 声卡的功能、技术指标与分类

### 1. 声卡的功能

音频卡也叫声卡，它的功能主要有以下几点。

#### （1）录制、编辑和回放数字声音文件

声音源可以是话筒，收录机或者光盘等，在声音处理软件的控制下，经过声卡采样，数

字化成数字语音文件，并可以播放这些文件或对它们进行编辑。不同音频卡和软件驱动程序录制的语音文件格式可能不同，但它们之间可以相互转换。例如，Creative Labs 用 VOC 作数字语音文件的扩展名，而在 Microsoft 的 Windows 下则以 WAV 为扩展名，它们之间就可以相互转换。WAV 文件通过 OLE (Object Linking and Embedding) 加入到其他 Windows 应用程序中。

通常音频录放采用：

- 数字化音频采样频率范围：5kHz~44.1kHz。  
量化位：8 位/16 位。  
通道数：立体声/单声道。
- 基本编码方法：PCM  
压缩编码方法：ADPCM (8:4, 8:3, 8:2, 16:4)  
CCITT A 一律 (13: 8)  
CCITT  $\mu$  一律 (14: 8)  
实时硬件压缩/软件压缩。
- 音频录放的自动动态滤波。
- 录音声源：麦克风、立体声线路输入、CD 输入可选麦克风自动增益控制 (AGC) 放大器，以适应麦克风灵敏度 (10~100mV)。
- 输出功率放大器，直接驱动扬声器，且输出音量可调 (0~15 级或手动调节)。

(2) 控制声音源的音量，混合后再数字化

音频卡驱动程序中通常有 Mixer 程序，用来控制音频卡上的混合器。

(3) 记录和回放数字声音文件时进行压缩和解压缩以节省存储语音文件的磁盘空间

对于立体声，如果不进行压缩的话，则数字化播放每分钟的数据量要求占据 10MB 的磁盘空间，即使是单声道，也不会少于 1MB。常见的自适应脉冲编码调制 (ADPCM) 压缩方法一般能得到 2:1 的压缩比而不会明显失真。

(4) 文语转换与语音识别

有些音频卡在出售时还捆绑了文语转换软件和语音识别软件。

1) 文语转换软件。文语转换 (text to speech) 就是把计算机内的文本转换成声音。一般音频卡都提供英语文语转换软件，如 Sound Blaster。

2) 语音识别软件。有的音频卡提供语音识别软件，如 Sound Blaster 卡上的 Voice Assist、Microsoft Sound System 卡上的 Voice Pilot 软件。这两个软件都是特定人的命令识别系统。通过软件可以利用语音来控制计算机或执行 Windows 命令。

(5) MIDI 接口和音乐合成

MIDI 规定了电子乐器与计算机之间相互数据通信的协议。通过软件，计算机可以直接对外部电子乐器进行控制和操作。

音乐合成功能和性能依赖于合成芯片。目前 Yamaha 的合成器芯片占有率最高，其中主要是 FMOPL 系列。这种芯片采用调频 (FM) 方式合成音乐，有的音频卡带有波形表音乐合成。如 Sound Blaster Ave32 采用 E-mu 的 EMU800 数字信号处理器，达到真实乐器效果的 CD 质量音响，它支持 32 复音的多音色 MIDI 通道。

音频卡的其他功能接口还有：

- CD-ROM 接口。目前音频卡的 CD-ROM 接口有多种。如 Sound Blaster 专用 CD-ROM 接口；Sony、Mitsumi CD-ROM 接口；SCSI-I 标准 CD-ROM 接口。
- 游戏棒接口。标准的 PC 游戏棒接口，可接一个或两个游戏棒。

## 2. 声卡的技术指标

1987 年，PC 机上诞生了第一块声卡——AdLib 卡（港台地区称之为魔音卡）。1989 年新加坡的 Creative Labs 生产的第一代 Sound Blaster（声霸卡）问世，很快取代 AdLib 成为 PC 机上的声音标准。由于 Sound Blaster 有较强的记录和播放数字化声音的能力，特别是其后续的版本提供了 MIDI 接口、立体声等新功能，一些商业应用软件也开始使用数字化声音和音乐。现在 Creative Labs 已经成为世界主要的多媒体配件生产厂家之一，其产品包括从低档的游戏卡到尖端的视频会议系统，应用十分广泛。

如果要求较好的演奏音响效果及专业编曲能力，普通声卡是不能胜任的，但对于商业性 OLE 应用或简单的作曲则绰绰有余。使用音效卡可播放预先录制的声音或音乐文件、产生丰富的声音效果、播放 CD 音乐等等。与 CD-ROM 驱动器相比，声卡种类更加繁多、价格品质相差悬殊，选择的余地较大，同时需要考虑的因素也较多。

### （1）采样率与量化位

衡量声卡录制和重放声音质量的主要参数是采样率与量化位（也称为分辨率或解析度），采样率与量化位越大，录制和重放声音质量与原始声音就越接近。在 MPC Level 1 的基本规范中将声卡量化位定为 8 位，但在 Level 2 中已将这一指标提升至 16 位（CD 音乐的水准）。

现在一般的声卡都可以在 44.1kHz 采样频率下对立体声源进行 16 位数字化录音（即每秒取样 44100 个 16 位双声道）和重放，但并不是每一种声卡记录和回放的质量都一样，声卡上的模拟/数字转换、前置放大器和数字电路的质量对声音质量的影响是非常大的，所以不能仅仅从广告介绍的性能指标来衡量声卡的质量，最好是亲自动手测试一下。

### （2）FM 合成与波形表

大多数普及型声卡采用 FM 合成法（与家用电子琴类似），即通过正弦波相互调制来模拟真实的乐器声音。这种方法成本较低，但也导致了在游戏或音乐演奏中产生的音效与实际的乐器明显不同，你只要实际听一下由 FM 合成的钢琴声就不难体会到这种差别。现今声卡的 FM 合成通常是使用日本 Yamaha 公司生产的 OPL-2（老式声卡上的芯片，也叫做 M3812，可合成 11 种单声道的声音）或 OPL-3（也叫做 YMF262，可合成 11 种单声道的声音）合成芯片。

较好的声卡采用的是波形表（Wavetable）合成技术来实现音乐合成（即所谓的波表卡）。波形表包含有真实乐器声音波形的数字记录，在演奏时将相应乐器的波形记录播放出来。显然，这种方法可以产生更加丰富逼真的音频和音乐。波形声音一般存储在声卡的 ROM 芯片中，但存储各种乐器声音记录需要大量的存储器（通常要有 2~4MB），这是造成波表卡价格较贵的主要原因。为了降低造价，有些声卡将 4MB 的波形表声音数据压缩为 2MB 以下，也有些卡使用硬盘来存放波形表，在需要时通过卡上的 RAM 来转存硬盘上的声音信息。

为了与原有的 FM 合成声卡的兼容性，波表卡上的合成芯片能完成 FM 合成的所有功能，如 Yamaha 公司非常流行的 OPL-4（可运行为较早的 OPL-2 和 OPL-3 芯片编写的所有程序）

就是典型的波表合成芯片。

如果经费不允许购买波表卡，可以先买一个带波表升级的 FM 合成声卡，待以后波表卡降价再将其升级为波表卡。进行波表升级时要将一块子板连接到声卡上，因此在声卡相邻的插槽上最好不要有其它的插件板，以免产生干扰或发生其它意外。目前国内市场上出售的许多声卡都有波表升级插口，但用于升级的波表子板却十分少见，笔者见到的只有 Creative Labs 的 Wave Blaster 等极少数几种，但随着多媒体应用的深入，这种情况会有变化。Wave Blaster 可以插在 Sound Blaster 16 系列声卡和部分兼容卡上，但是价格相当可观。

### (3) 兼容性

声卡的兼容性是购买时应考虑的重点。如果你计划只在 Windows 下使用声卡，那么可以忽略其与 Sound Blaster 和 AdLib 之类卡的兼容性，只要该卡能提供适当的 Window 声音驱动程序就行了。但如果你想在 DOS 环境下使用，特别是想用来玩游戏，那么就要注重声卡与 Sound Blaster 和 AdLib 的兼容性，只有这样才能使用市场上绝大多数基于 DOS 的游戏软件。由于一些兼容声卡生产厂家并无 Creative Labs 的技术许可，可能不会像其声称的那样与 Sound Blaster 卡 100% 兼容。

兼容性较好的声卡一般可以完全兼容如下几个标准：

- AdLib
- Sound Blaster 和 Sound Blaster Pro
- Microsoft Windows Sound System 2.0
- MPC Level 2

### (4) 外围接口

声卡上一般都有与其他设备的接口部件，这包括 MIDI/GAME 端口、I/O 端口、CD-ROM 端口等。但接口部件的类型和质量是因卡不同而相异的。所以在选择声卡时，应该根据需要进行选择。例如应该选择符合一定标准的 MIDI 接口；如果你打算同时使用 CD-ROM 驱动器的话，最好购买带有适当 CD-ROM 接口的声卡，这样既可以节约资金，又能省下一个扩展槽。

### (5) 音频压缩

高质量的数字化音频会带来存储空间的困难，尽管不像视频那么严重。例如，1min CD 音质的未压缩立体声音频信号就要占用约 11MB 的硬盘空间。因此，声卡应支持几种语音压缩标准，其中主要有 ADPCM（自适应差分脉冲编码调制）和 ACM（微软音频压缩管理器，Microsoft's Audio Compression Manager）等，压缩比约为 4:1~6:1。

### (6) DSP 芯片

一些较高档的声卡（如 Sound Blaster 16 ASP）带有数字信号处理器（DSP）芯片，这是一个专用的数据处理器，可以通过软件编程来实施音频处理和压缩等任务，从而减轻了 CPU 的压力，因为在普通声卡上，这些任务都是由 CPU 完成的。如果你的系统速度较低（如 386 SX），那么 DSP 芯片可能是极其重要的。但真正重要的是 DSP 芯片能够编程以进行其它的处理，如为声卡增加传真、调制/解调功能或进行视频处理等等。目前关于 DSP 芯片的使用还未形成一致的标准，对此普通用户不必过多考虑，如无特殊需要，没有 DSP 芯片也能应付。



### (7) 软件支持

声卡必须具备的软件有 DOS 和 Windows 的驱动程序, 软件混频器和 CD 唱盘播放程序等。有的声卡(如 Media Magic 等)还包括由 Voyetra Technologies 提供的非常好的商业软件, 有像家用音响似的 AudioStation、进行 MIDI 创作的 Orchestrator、数字化录音编辑工具 WinDAT 和简单的多媒体写作工具 SoundScript 等等。

从一开始, Creative Labs 的 Sound Blaster 系列就以其丰富的软件支持闻名于世, 如 QSound、TextAssist、VoiceAssist、Talking Scheduler、WaveStudio 和 FM Intelinent Organ 等都是非常有名的软件。

## 3. 声卡的分类

### (1) 按应用环境分类

按照声卡的应用环境, 声卡基本可以分为 DOS/GAME 和 Windows 两种环境。这两种声卡分别以 Sound Blaster 和 Windows Sound System 为代表。前者, 即 Sound Blaster 是 GAME 声卡的事实标准, 几乎所有的 DOS 环境下的游戏都支持 Sound Blaster。而在 Windows 环境下, Windows Sound System 无疑就是标准, 它以多媒体计算机为背景, 由 Microsoft 公司提出, 目的是为了统一声卡的标准, 最终为应用提供方便。

### (2) 从声卡的技术角度分类

从声卡所采用的技术上来看, 声卡主要可分为三类: 一是 DSP 技术为基础的声卡, 这种声卡本身带有一个 DSP 处理器, 所以大部分的控制功能都可以不依赖主机而自行完成, 与主机之间只需进行数据交换。如 Creative 公司的 Sound Blaster 系列、维用公司的 adsp 系列。二是全硬件声卡, 这种声卡采用称为 CODEC 的芯片, 控制声音的采样与播放, 而其它的控制全部依赖于主机, 占用较多的主机时间, 但成本却很低。如 Windows Sound System。三是结合一类和二类两种声卡的优点, 采用有限可编程控制器, 使声卡具有一定能力的自管理功能, 又不致于成本太高、复杂, 这两种声卡, 如 Turtle Beach 公司的 Mad 16 等, 它们以硬件模拟 DSP 的某些功能, 以减轻主机的负担, 同时又保持低成本, 省去 DSP 的开销。

### (3) 根据总线的不同分类

根据总线的不同, 我们把声卡分为两大类, 一种是 ISA 声卡, 另一种是 PCI 声卡, 由于两种端口不能互相通用, 因此我们在安插声卡时不能插错。主板上的 ISA 插槽是黑色的, 比 PCI 槽长, 其中的金属簧片也比 PCI 的宽; PCI 插槽呈白色, 相对较短, 其中的簧片很细, 分布密集。

由于 PCI 总线的优越性, PCI 声卡有着许多 ISA 声卡无法拥有的特性, 但这并不是说 PCI 声卡的音质一定比 ISA 好, 决定音质的好坏主要由声音处理芯片、MIDI 的合成方式和制造工艺等, 并不仅仅是总线的不同。

当然还可以按照声卡的组成结构, 分为普通声卡和集成主板的声卡。按照声卡取样分辨率的位数不同, 可分为 8 位声卡、准 16 位声卡、真 16 位声卡、32 位声卡等等。按照声卡功能的不同, 可分为单声道声卡、真立体声声卡、准立体声卡等等。

## 2.2.2 声卡的组成和布局

一块典型的声卡一般都有 MIDI/GAME 端口, 除此之外, 声卡通常还包含有一个麦克风

输入端口、一个用于采集来自 CD 唱机或盒式录音机信号的线入端口（Line-in）、一个耳机插口和通过标准音响系统或放大器重放声音的线出端口（Line-out），声音处理芯片，功率放大芯片。此外，声卡上还应提供有 CD-ROM 驱动器重放 CD 唱盘的音频连接器，这样 CD 唱盘上的声音可以与来自其他声源的信号混合，经声卡上的混频器进行同步输出。图 2-2 是一个典型声卡的布局图。

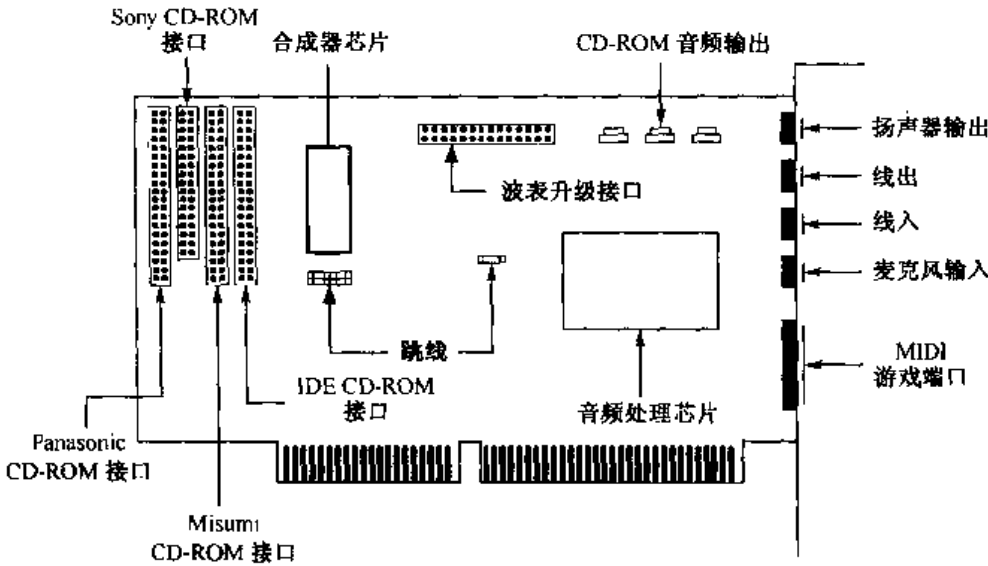


图 2-2 典型声卡的平面图

下面对上述图中的几个地方作一些解释。

1. MIDI/GAME 端口

因为 MIDI（音乐设备数字接口）也是一种数字化声音，它不是由实际声音进行数字化而产生的。在 MIDI 文件中含有播放某些乐器声音的指令和要产生的效果，因此符合 MIDI 标准的任何附加设备（不仅仅是电脑，还有如键盘合成器和吉他等等）都能生成和播放 MIDI 文件。通过声卡上的 MIDI 接口可以连接其它 MIDI 设备，构成以电脑为核心的个人音乐作曲和演奏平台。

MIDI 接口所支持的标准很多，除基本的 MIDI 外，还有 MPU-401、General MIDI 等等，视声卡上的合成器芯片和接口而定。除早期生产的卡外，目前的声卡一般都能支持 MPU-401/UART 和 General MIDI 两个标准。

MIDI 标准规定了输入/输出通道（称为 MIDI 端口）类型、连接电缆及插座的形式等。MIDI 端口有 MIDI In（接收数据）、MIDI Out（发送数据）和 MIDI Thru（转送数据）三种类型。

General MIDI（通用 MIDI）标准是乐器制造商制订的标准，它规定了不同的乐器声音放在合成器存储器的特定位置，从而保证能逼真地重现该乐器的声响。与 General MIDI 标准的兼容性确保声卡能重放大型商用音乐文件库中的音乐。表 2-1 列出了微机上几种经常使用的 MIDI 标准及其特点。

表 2-1 微机上使用的 MIDI 标准

MIDI 标准	描 述
MIDI	音乐设备数字接口
General MIDI	规定了 MIDI 文件中乐器声音的排列顺序
MPU-401	Roland 公司制订的 MIDI 标准，为音乐界和乐器制造商所采用
MT-32	与 General MIDI 标准类似，但乐器声音排列顺序略有不同

声卡上的 MIDI 端口通常与游戏端口是合一的，因此这两个功能不能同时使用（如要同时使用需用专门的转换盒）。在不使用外部 MIDI 设备时，这个端口可以连上一个游戏杆（Joystick），对于诸如飞行模拟或探险类的游戏非常方便。有些声卡可用软件方法设置 MIDI/GAME 端口的使能，而其它的卡则需要改变跳线才能切换。图 2-3 所示为 MIDI 及游戏摇杆接口。

## 2. I/O 端口

关于 I/O 端口的连接情况，图 2-4 画出了声卡 I/O 端口分布情况。

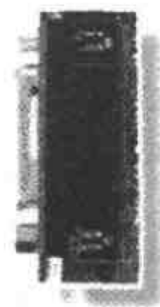


图 2-3 MIDI 及游戏摇杆接口

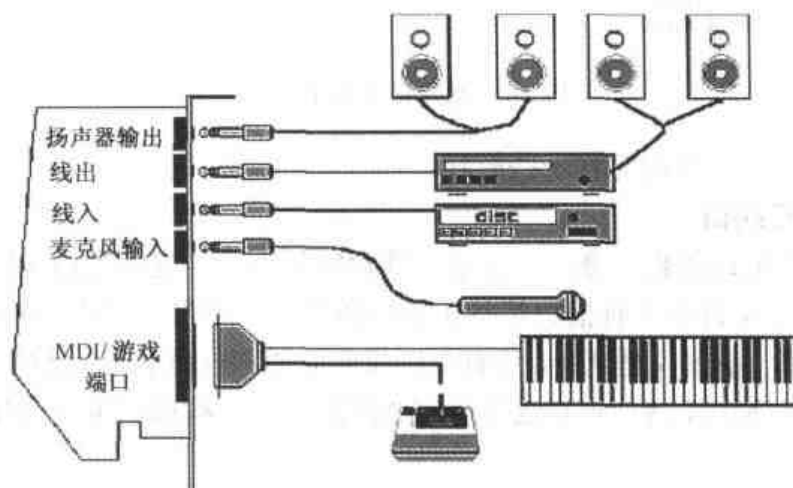


图 2-4 声卡的 I/O 端口

## 3. CD-ROM 接口

许多声卡都提供了常用的 Mitsumi、Panasonic 和 Sony 三种 At-Bus 接口，少数声卡也带有 IDE 接口，可以满足基本需求。这些接口即使闲置不用，也不会影响声卡的正常使用。Mitsumi、Panasonic 和 IDE 接口的形式一样，要注意识别，而 Sony 接口较短，不会与其他接口混淆。

## 4. 声音处理芯片

通常是最大的四边都有引线的那只集成块，上面标有商标、型号、生产日期、编号、生产厂商等重要信息。声音处理芯片基本上决定了声卡的性能和档次，其基本功能包括对声波采样和回放的控制、处理 MIDI 指令等，有的厂家还加进了混响、合声、音场调整等功能。

声卡上声音处理芯片有的可能是 3~6 块 IC 构成的芯片组。AC97 规范为了保证声卡的信噪比（SNR）能够达到 80dB（分贝）以上，要求声卡上的 ADC、DAC 处理芯片与数字

音效芯片分离，因此，高档声卡上的芯片一般不止一块。

世界上主要的声音处理芯片有 SB、ESS、OPTI、AD、YMF、ALS、ES、S3、AU 等，而目前在声卡界居于领头羊位置的则是 Creative 和 Diamond。

### 5. 功率放大芯片

从声音处理芯片出来的信号还不能直接推动喇叭放出声音，绝大多数声卡都带有功率放大芯片（简称功放）以实现这一功能。声卡上的功放型号多为 XX2025，功率为 2×2W，音质一般。由于它在放大声音、音乐等信号的过程中也同时放大了噪声信号，所以从其输出端（Speaker Out）输出的噪声较大。这个缺点在前两年重视功能的潮流中显得并不突出，但是现在人们对音质的要求越来越高，于是就有厂商想出了一些改进的方法，主要是在功放前端加入滤波器来滤掉一些高频的噪声信号，可是这样一来也滤掉了很多高频的音乐信号。其实，指望声卡上的功放芯片能带来良好的音质是不现实的，一个比较好的解决方法是绕过功放，利用声卡上线路输出（Line Out）端口连接音响，这样，音质的好坏就直接取决于声音处理芯片和外接的音响设备（一般是有源音箱）的档次了。

### 6. 跳线和 SB—Link 接口

在较早期面市的 ISA 声卡上多数都有跳线，它的作用是给 ISA 声卡设置通道和中断信号（DMA 和 IRQ）以使操作系统与声卡能进行信号传输。现在的绝大多数声卡采用了软件设置通道的方式，但是其上还是有跳线，这种跳线的作用是区分输出端的那个插孔是“Line Out”还是“Speaker Out”。PCI 声卡符合 PnP（即插即用）原则，它不需要设定通道，因此与 DOS 应用程序有兼容性问题，造成 DOS 游戏有时不能发声或发声不正常，为解决这个问题，大多数 PCI 声卡都有一个与主板 SB—Link 接口相连接的插座（连线随声卡配置），在 DOS 下强制分配通道以解决兼容性问题。

## 2.2.3 声卡的工作原理

开发生产音频卡的公司很多，其中最有影响的公司是新加坡创新科技有限公司（Creative Labs. Inc.）开发的系列产品 Sound Blaster 系列音频卡，它是集语音与音乐于一体的多媒体音频卡，它不但具有优良稳定的硬件特性，而且还有丰富的软件。尽管目前世界各国开发了很多品牌的音频卡，但大多数都声明与 Sound Blaster 兼容，因此它已成为多媒体计算机公认的音频接口标准。目前 Creative 公司推出的音频卡的型号有 Sound Blaster, Sound Blaster 16, Sound Blaster AWE 32, Sound Blaster AWE64, Sound Blaster Live! 及 Sound Blaster Live! Value。Sound Blaster Live! 是 Creative 公司推出的环境音响效果新平台，通过环境音效技术，进行环绕音响渲染，产生栩栩如生、身临其境的音响效果，其动感效果有如置身于乐队之中。1998 年推出的 Sound Blaster Live! Value，通过专业合成器和数字 I/O，可以提供超级影院的音频保真度，获得平均为 120dB 的低噪声水平。其他特性如下：

#### （1）强劲的音频处理引擎

EMU10K1 音频处理器是目前功能最强大集成的音乐、音频及效果引擎。每个信号都能在 32 位，192dB 和 48kHz 模式下通过 8 点插值处理进行声音的平滑，并实时提高所有音源的质量。它还提供超越昂贵的专业效果设备的实时效果，如混音、和声、滑音、回声和变调等。

#### （2）环境音效增加现有音频的内容

体验超越现有 3D 音频，更具感染力、更理想的音频效果。通过在游戏或应用软件中加

入预置的类似大厅、洞穴、水下等环境，将使用户体验到那令人吃惊的逼真效果。游戏环境设置还能瞬间使游戏变得栩栩如生。

(3) Sound Blaster™PCI 标准

在享受 PCI 总线优越性的同时，将确保获得对 PCI 总线近乎完美的 Sound Blaster™ 兼容性。它支持所有 MS-DOS 和 Windows 程序，可完全取代现有的 ISA 卡。

(4) 多音箱输出

内置支持 2~4 个模拟音箱，使 PC 获得环绕立体声效果。

(5) 256 复音音乐合成器

256 复音的品质和性能已经远远超越了目前大多数专业音乐设备。使用系统内存，可以选择 E-mu 的 2MB、4MB 或 8MB SoundFont®专业音色库进行音乐创作，或使用多达 32MB 的内存以获得极高的 SoundFont 音乐保真度。所有这些都不需牺牲 CPU 的利用率。

(6) 环境音效功能扩展集/广泛的软件支持

环境音效功能扩展集 (EAX™) 是一个公开的标准，它已获得软件开发商最广泛的支持。支持 Microsoft DirectSound®, DirectSound®3D 及它们的衍生技术。具有先进的结构以支持即将到来的如 WDM, DLS 和 IEEE1394 等设备。

音频卡的工作原理如图 2-5 所示，主要组成部分如下。

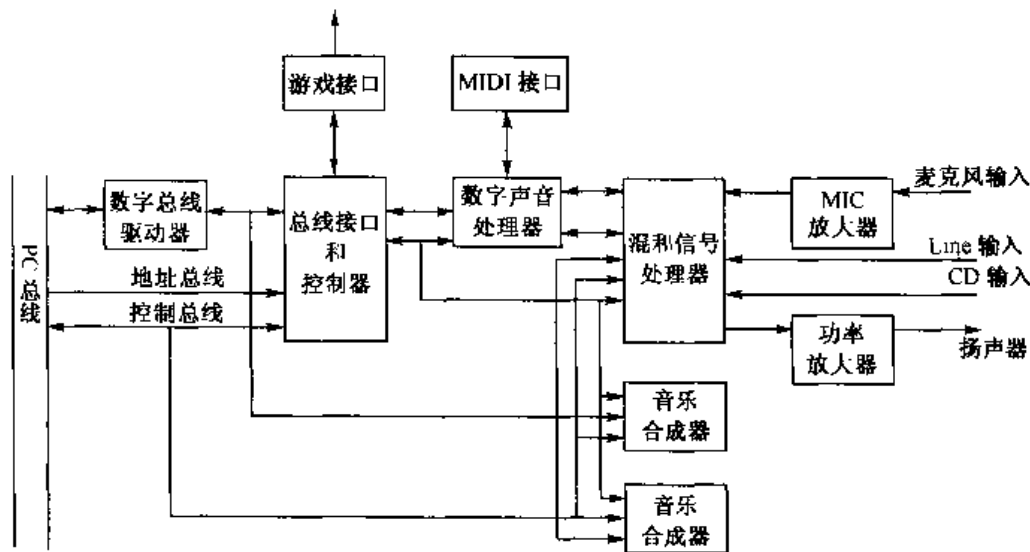


图 2-5 声卡原理框图

声音的合成与处理：这部分是音频卡的核心，它一般由数字声音处理器 (digital sound processor)、FM 音乐合成器及 MIDI 接口控制器组成。它的主要任务是完成声波信号的模/数、数/模转换，利用调频技术控制声音的音调、音色和幅度。FM 的音乐合成器具有 11 个复音 4 操作器或 20 复音 2 操作器的功能。

混合信号处理器及功率放大器：内置数字/模拟混音器，混音器的声源可以是 MIDI 信号，CD 音频，线性输入、话筒和 PC 机的扬声器等，可以选择输入一个声源或将几个不同声源进行混合录音。立体声数字化声道，可编程设定 16 位或 8 位数字化立体声或单声道模式，可编程设定采样频率，其范围从 5kHz~44.1kHz 之间线性分布；使用高、低 DMA 通道进行录音和放音；可选用动态滤波器进行数字化音频录音和回放。麦克风输入和扬声器输出

都有功率放大器。输出每一声道具有  $4\Omega$ 、 $4W$  的输出功率，也可外接音频放大器。对于主音量，MIDI 设备的数字化音频、CD 音频、线路输入，话筒和计算机的扬声器等，可以通过软件控制其音量，对于所有音源可分为 32 级音量控制，每级相差 1.5db。

计算机总线接口和控制器：早期的音频卡是 ISA 总线接口，近期音频卡是 PCI 总线接口。总线接口和控制器是由数据总线双向驱动器、总线接口控制逻辑、总线中断逻辑及 DMA 控制逻辑组成。总线接口控制逻辑包括：地址比较选中、地址选通、数据选通、读/写信号，地址锁存、总线使能以及总线错误等信号。音频卡可以跨接线设定基本 I/O 地址、中断向量（IRQ）、DMA 通道三个参数，避免与主机其他板卡冲突。

## 2.2.4 SPDIF 数字音频接口

虽然目前很多高档声卡采用了各种有益的方法来提高自身的音质表现，但是电脑机箱内复杂的电磁干扰依然是难以避免的。那么如何进一步提高声卡的音质呢？SPDIF 接口技术为我们提供了一个很好的解决方案！

### 1. SPDIF 概述

SPDIF 是 SONY、PHILIPS 数字音频接口的简称。就传输方式而言，SPDIF 分为输出（SPDIF OUT）和输入（SPDIF IN）两种。目前大多数的声卡芯片都能够支持 SPDIF OUT，但我们需要注意的是，并不是每一种产品都会提供数码接口。譬如早期的一些中高档 YAMAHA 724 声卡（如中凌雷公 724、YAMAHA 原厂的 WF192D）普遍含有一个 SPDIF OUT，而一些中小厂的廉价产品就不提供这个接口。支持 SPDIF IN 的声卡芯片相对少一些，如 EMU10K1、YMF-744 和 FM801-AU、CMI8738 等。SPDIF IN 在声卡上的典型应用就是 CD SPDIF，但也并不是每一种支持 SPDIF IN 的声卡都提供这个接口。就传输载体而言，SPDIF 又分为同轴（图 2-6 是典型的同轴 SPDIF 接口）和光纤两种（图 2-7 是典型的光纤 SPDIF 接口），其实它们可传输的信号是相同的，只不过是载体不同，接口和连线外观也有差异。但光信号传输是今后流行的趋势，其主要优势在于无需考虑接口电平及阻抗问题，接口灵活且抗干扰能力更强。通过 SPDIF 接口传输数码声音信号已经成为了新一代 PCI 声卡普遍拥有的特点。

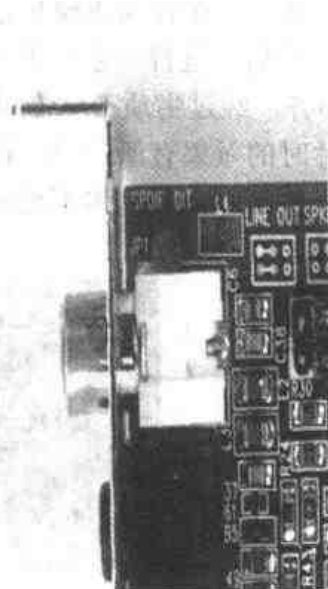


图 2-6 典型的同轴 SPDIF 接口

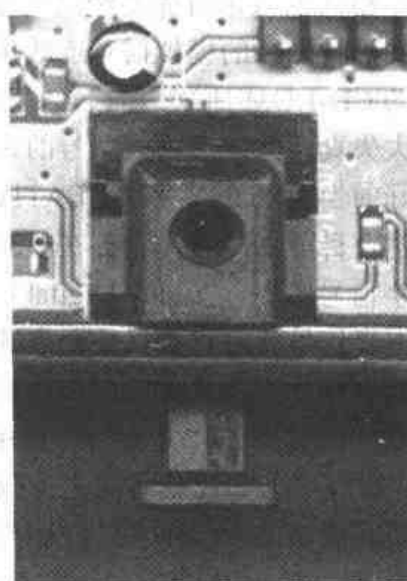


图 2-7 典型的光纤 SPDIF 接口

## 2. SPDIF 在多媒体声卡上应用的优势和不足

在目前的家用多媒体声卡上, SPDIF 同轴电信号输出主要用来传输 Dolby Digital AC-3 信号和连接纯数字音箱。光纤输出则主要用来连接 MD 等数码音频设备, 以实现几乎无损的音频录制。SPDIF IN 主要应用于传输数字 CD 信号, 也就是让电脑以数字方式播放唱片。下面笔者对大家比较疑惑的问题作一些说明。

### (1) SPDIF 是传输通道

需要特别解释的是, 大家不要以为使用 SPDIF 传输 AC-3 信号就是 AC-3 解码, 目前民用声卡中还没有一款产品能够支持硬件等级的 Dolby Digital 解码, SPDIF 在此时的功能主要是把数字 AC-3 信号从声卡传输到解码器。

### (2) 数字音箱与数字声卡的关系

前面我们就提到过, 声卡的数字模拟转换工作是交给 CODEC 芯片来完成的。但是我们的电脑机箱内依然存在着严重的电磁波, D/A、A/D 转换仍然会受到比较严重的信号干扰。许多专业音频录音卡普遍采用将 CODEC 外置的做法, 把数模转换部分以及各类外部接口等单独做成一个外置盒, 以提高音质。但是这样做的直接后果便是成本大幅度提高, 在家用多媒体市场肯定是曲高和寡的。那到底有没有价廉物美的办法呢? 一些音箱厂家就想出了把 D/A 转换工作从声卡上转移到音箱上的方案, 数字式多媒体音箱也就应运而生了, Creative 的 FPS2000 Digital、Sound Works 2.1Digital 就属于这种类型。这种方案的基本原理就是声音信号不经过声卡 CODEC 芯片的转换处理, 直接以 PCM 格式, 使用声卡上的同轴 SPDIF OUT, 以纯数字方式传输到数字音箱中, 通过音箱内置的 D/A 转换器解码, 随后放大输出。这样干扰减小了, 信噪比自然有所提高。然而目前主要不足之处在于, 眼下部分数字音箱的 D/A 转换单元、放大器、扬声器质量不高, 造成数字式传输的优势不能被完美的表现出来。

### (3) 唱片数字式播放的问题

Creative SB Live! 声卡上率先拥有一个两针的 CD SPDIF 接口 (图 2-8 所示为典型 CD SPDIF 接口), 从传输形式来看, 它属于 SPDIF IN 的范畴。那么它存在的意义又是什么呢? 我们都知道 CD 唱片上的声音信息是用数字“0”和“1”来表示的。以往 CD-ROM 在播放唱片时, 数字格式的音乐首先要经过光驱内部的 D/A 处理。在转换成模拟信号后, 经过我们时常使用的那种四针的模拟信号连线传输到声卡上, 然后再进行一系列处理。问题的关键在于, 不同的 CD-ROM 所采用的 D/A 芯片质量参差不齐, 经过劣质 D/A 转换后输出的模拟信号存在很大失真, 造成了不同型号的光驱在播放唱片时的效果有所差异。在 CD 解码质量上口碑比较好的当属 SONY 和 CREATIVE 的产品, 一些杂牌光驱则惨不忍听。为了避免这种问题的发生, 目前大多数光驱都在模拟信号输出插针的旁边加上了数字信号输出 (Audio Digital)。通过这个两针的接口, 唱片声音信号就可直接以数码方式传输到声卡上, 将 D/A 转换交给音频处理芯片来完成, 而前提则必须是声卡芯片可以完成相关的转换工作并支持 SPDIF IN, 能够接收数字信号。CD 播放的信噪比就将随之大幅度提升。目前可以支持 CD SPDIF 接口的声卡不多, 在国内市场上可以买的只有 SB Live! 系列、黑金 FM801 和一些 CMI8738 声卡。遗憾的是, 一些光驱虽然拥有数字音频信号输出功能, 但问题多



图 2-8 典型 CD SPDIF 接口

多,有的是无法正常出声,有的则反映杂音不断。究其原因可能有两点:光驱的数字输出口不标准或根本形同虚设;SB Live!声卡对光驱数字输出口的兼容性不佳。最近又有一个关于数字式唱片播放的热点来自 Microsoft 发布的 Windows MediaPlay 7.0。在这个微软最新推出的多功能音频播放软件中集成了数字 CD 播放功能,而且它宣称无论你的声卡是否支持 SPDIF IN,光驱是否带有数字音频信号输出,唱片都可以以数码方式播放。这究竟是怎么回事儿呢?其实 Windows MediaPlay 7.0 是通过光驱的 IDE 数据线将数字音频信号传输到系统总线,所以播放 CD 的时候数据灯会闪亮。最后再通过声卡进行 D/A 解码并输出,这种方式会增加系统负担,但不是很明显。

## 2.2.5 音频卡的发展和改进

MPC3 对音频卡的要求是目前对音频卡的标准要求,但这还远远没有达到音频卡发展的目标,音频卡在近期的发展将主要集中在进一步改善声音质量、统一音频卡标准、简化安装方法、三维环绕立体声、全双工声音处理、与通信技术的结合以及单一芯片等方面。

### 1. 改善声音质量

MPC3 没有对音频卡的信噪比作出明确的规定,但音频卡的信噪比在很大程度上决定了其音质,所以音频卡应具有优良的信噪比。优秀的音频卡都将在其 DSP 中增加动态滤波去噪功能,在录音时能有效地去除杂音。同时音频卡的采样大小现已逐步向 64-bit 方向发展,这可大大提高声音的质量。

### 2. 统一音频卡标准

只有在统一的标准下或兼容主要的音频卡标准才能适应尽可能多的应用软件。世界上主要的音频卡软件标准有 Adlib, Sound Blaster Pro, Microsoft Sound System, Roland MPU401 和 MT32 等,在一时难以形成统一标准的情况下,音频卡应能兼容这些音频卡标准,才能适应不同软件环境的需要。

### 3. 简化安装的即插即用音频卡

现在音频卡的安装往往要涉及 I/O 端口基地址、IRQ 号、DMA 通道号等复杂的设置,这对普通用户来说显得太难了,许多计算机用户认为音频卡是现代计算机中最难设置的设备,更糟糕的是,基地址、I/O 端口、IRQ 号、DMA 通道号之间的冲突将引起系统严重错误或死机,甚至系统崩溃。

为了解决外部设备之间 I/O 端口、IRQ 号、DMA 通道号设置的冲突,Microsoft 和 Intel 联合制定了 ISA 总线的即插即用标准。符合即插即用方式的外设将不再需要用户自己设置,系统将自动地从可用的系统资源中为设备分配 I/O 地址、IRQ 号、DMA 通道号等。理论上来说,操作系统可从主板及其上的 BIOS 中得到正确的 I/O 端口、IRQ 号、DMA 通道号服务,任何符合 PnP 标准的附加卡都可自行安全地安装。这要求操作系统能识别硬件并自动地安装驱动程序。Microsoft Windows95 是支持 PnP 的操作系统,Windows NT 和 IBM OS/2 将在新版中增加 PnP 特征。事实上,市面上有一些应用软件可以使 Windows3.1 具备 PnP 功能。

虽然因为即插即用标准推出的时间较短,现有许多软件并不能很好地识别即插即用音频卡,甚至不支持 PnP 的操作系统如 Windows NT、IBM OS/2 上安装即插即用音频卡可能会出现许多问题,但易于安装的音频卡终究更易于被用户所接受,安装简便的即插即用音频卡将是音频卡的重要发展方向。



#### 4. 三维环绕立体声

三维环绕(3D)立体声是现代音频卡的另一重要特征,它使音频卡具有更逼真的音响效果,让用户在使用计算机时,特别是在玩游戏时,感觉到声音来自各个不同的方向。这种逼真的三维环绕立体声效果能给用户提供一个真实可信的游戏经历,产生身临其境的感觉。

音频卡产生 3D 声音效果主要基于两种产生 3D 声音的技术,即 Qsound Labs Inc. 的 Qsound 技术和 SRS Labs Inc. 的 SRS (Sound Retrieve System) 技术。这两家公司都已生产大量的 3D 声音处理器芯片。只需要两只普通的音箱,3D 音频卡就能使 PC 机处理声音,并将其映射到环绕用户的所有方向上。它能模拟各种真实的声音效果,如在玩射击活动物体的游戏时,用户能感觉到射击来自不同的方向并消失于其他方向。

SRS 技术是在普通立体声的基础上,使用硬件电路和软件共同控制、共同作用的方法,使其在空间交叉形成多个虚拟音源,模拟了声音在自然界中的传输效果,在大自然中,声音是通过空气传播的,并具有各种吸收反射、漫反射等过程,所以在自然界中较空旷的地方,如音乐厅、大厅、峡谷或较空旷的室内,声音的效果就比较好,而 3D SRS 系统就实现了这种音响效果,此技术以前只是应用在高级音响中,现在已经应用于 PC 机使用的普通音频卡上,并在价格和效果上使普通用户可以接受。3D SRS 通过硬件和软件综合控制,硬件电路较成熟,软件控制简便。

Qsound 是通过软件实现的,在使用音效时需加载 Qsound 专用处理软件。其原理是通过软件控制左右音箱的发声过程,可形成 180° 的音场,从而形成空间立体环绕声效果。此种效果的一个明显缺点是必须加载软件,声音效果通过对音箱的左右音箱控制的时间顺序的变换,形成左右声道的声音的变换,从而形成左右声道声音之间的流动效果。

#### 5. 全双工声音处理

全双工音频卡能同时进行放音和录音,这是音频卡的另一个发展趋势。全双工特性主要应用于最新的国际互联网络 Internet 的语音通话应用(即 Internet 电话)中。Internet 电话允许两个用户使用语音设备通过 Internet 连接进行通话。在连接时,音频卡录下用户的语音,Internet 软件把语音分成不同的信息包(分组报文),并通过 Internet 传送出去;对接受信息的用户方面来说,只需将 Internet 传来的信息包进行组装还原,并通过音频卡播放出来。

使用 Internet 电话的最大好处就是用户只需缴纳本地使用费就能与世界各地的人通话,而获得这种好处的要点就是安装能同时进行放音和录音的全双工音频卡。Aztech Nova Extra II 16-3D 是市面上最常见的全双工音频卡。

#### 6. 与通信技术的结合

在现代信息社会中,电子通信把整个世界联系在一起,在音频卡中加上数据 Modem、传真等通信功能是音频卡的发展趋势。Aztech Audio Telephony 2000 是这种与通信技术相结合的音频卡的典范,它是一种集六种功能于一体的多功能电信卡,具有数据 Modem、传真、自动回答电话、录音电话、16 位音频卡、带有 CD-ROM 驱动器接口等多种功能,可以说是集整个办公自动化所需设备于一卡之中。

#### 7. 单一芯片

现在音频卡处理数字声音的 DSP 芯片,合成声音的合成器芯片,需要两个芯片,将来的音频卡将朝着单一芯片方案发展,即将 DSP 芯片和合成器芯片等音频卡的主要组件集成到一块芯片中,进一步简化音频卡的电路,降低生产成本。1995 年 Yamaha 公司的 FM 专利

已过期,许多公司都已实现集成音频编码/解码器(比如 A/D、D/A 转换器)、混音芯片及 FM 合成器于单一芯片的目标。随着 DSP 和超大规模集成电路的发展,整块音频卡都将集成于一块芯片中,并成为计算机主板的一部分。

## 2.3 音频编码基础和标准

### 2.3.1 音频编码的基础

从信息保持的角度讲,只有当信源本身具有冗余度,才能对其进行压缩。根据统计分析结果,语音信号存在着多种冗余度,其最主要部分可以分别从时域和频域来考虑。另外由于语音主要是给人听的,所以考虑了人的听觉机理,也能对语音信号实行压缩。

#### 1. 时域信息的冗余度

##### (1) 幅度的非均匀分布

统计表明,语音中的小幅度样本比大幅度样本出现的概率要高。又由于通话中必然会有间隙,更出现了大量的低电平样本,此外,实际讲话信号功率电平也趋向于出现在编码范围的较低电平端。因此,语音信号取样值的幅度分布是非均匀的。

##### (2) 样本间的相关

对语音波形的分析表明,取样数据的最大相关性存在于邻近样本之间。当取样频率为 8kHz 时,相邻取样值间的相关系数大于 0.85;甚至在相距 10 个样本之间,还可有 0.3 左右的数量级。如果取样速率提高,样本间的相关性将更强。因而根据这种较强的一维相关性,利用 N 阶差分编码技术,可以进行有效的数据压缩。

##### (3) 周期之间的相关

语音信号虽与电视信号有许多相似之处,但其最大的不同是语音信号的直流分量并不占主要成分。因为光信号是非负的,而语音信号却可正可负。虽然语音信号需要一个电话通路提供整个 300~3400Hz 的带宽,但在特定的瞬间,某一声音却往往只是该频带内的少数频率成份在起作用。当声音中只存在少数几个频率时,就会像某些振荡波形一样,在周期与周期之间,存在着一定的相关性,利用语音周期之间信息冗余度的编码器,比仅仅只利用邻近样本间的相关性的编码器效果要好,但要复杂得多。

##### (4) 基音之间的相关

人的说话声音通常分为两种基本类型:

第一类称为浊音(voiced sound),由声带振动产生,每一次振动使一股空气从肺部流进声道,激励声道的各股空气之间的间隔称为音调间隔或基音周期。一般而言,浊音产生于发元音及发某些辅音的后面部分。

第二类称为清音(unvoiced sound),一般又分成摩擦音和破裂音两种。前者用空气通过声道的狭窄部分而产生的湍流作为音源;后者声道在瞬间闭合,然后在气压激迫下迅速地放开而产生了破裂音源。语音从这些音源产生,传过声道再从口鼻送出。清音比浊音具有更大的随机性。

浊音波形不仅显示出上述的周期之间的冗余度,而且还展示了对应于音调间隔的长期重复波形。因此,对语音浊音部分编码的最有效的方法之一是对一个音调间隔波形来编码,并以其

作为同样中其他基音段的模板。男、女的基音周期分别为 5~20ms 和 2.5~10ms，而典型的浊音约持续 100ms，一个单音中可能有 20~40 个音调周期。虽然音调周期间隔编码能大大降低码率，但是检测基音有时却十分困难。而如果对音调检测不准，便会产生奇怪的“非人音”。

### (5) 静止系数

两个人之间打电话，平均每人的讲话时间为通话总时间的一半，另一半时间听对方讲。听的时候一般不讲话，而即使是在讲话的时候，也会出现字、词、句之间的停顿。通过分析表明，话音间隙使得全双工话路的典型效率约为通话时间 40%（或静止系数为 0.6）。显然，话音间隔本身就是一种冗余，若能正确检测出该静止段，便可“插空”传输更多的信息。

### (6) 长时自相关函数

上述样本、周期间的一些相关性，都是在 20ms 时间间隔内进行统计的所谓短时自相关。如果在较长的时间间隔（比如几十秒）进行统计，便得到长时自相关函数。长时统计表明，8kHz 的取样语音的相邻样本间，平均相关系数高达 0.9。

## 2. 频域信息的冗余度

### (1) 非均匀的长时功率谱密度

在相当长的时间间隔内进行统计平均，可得到长时功率谱密度函数，其功率谱呈现强烈的非平坦性。从统计的观点看，这意味着没有充分利用给定的频段，或者说有着固有的冗余度。特别是，功率谱的高频能量较低，这恰好对应于时域上相邻样本间的相关性。此外，还可以看到，直流分量的能量并非最大。

### (2) 语音特有的短时功率谱密度

语音信号的短时功率谱，在某些频率上出现峰值，而在另一些频率上出现谷值。这些峰值频率，也就是能量较大的频率，通常称为共振峰频率。此频率不止一个，最主要的是第一和第二个，由它们决定了不同的语音特征。另外，整个谱也随频率的增加而递减。更重要的是，整个功率谱的细节以基音频率为基础，形成了高次谐波结构。这都与电视信号类似，仅有的差异在于直流分量较小。

## 3. 人的听觉感知机理

语音最终是给人听的，所以要充分利用人的听觉生理、心理特性对于语音感知的影响，以免做“即使记录了，人耳也听不见”的无用功。

### (1) 人的听觉具有掩蔽效应

当几个强弱不同的声音同时存在时，强声使弱声难以听见的现象称为同时掩蔽，它受掩蔽声音和被掩蔽声音之间的相对频率关系影响很大；声音在不同时间先后发生时，强声使其周围的弱声难以听见的现象称为异时掩蔽。

(2) 人耳对不同频段的声音的敏感程度不同，通常对低频端较之对高频端更敏感  
即使是对同样声压级的声音，人耳实际感觉到的音量也是随频率而变化的。

### (3) 人耳对语音信号的相位变化不敏感

人耳听不到或感知极不灵敏的声音分量都可视为冗余的。

音频编码的目的在于压缩数据。在多媒体音频数据的存储和传输中，数据压缩是必须的。通常数据压缩造成音频质量的下降、计算量的增加。因此，人们在实施数据压缩时，要在音频质量、数据量、计算量复杂度三方面进行综合考虑。

为了实现音频数据压缩，多方面的专家致力于算法的研究，众多的企业致力于芯片和产

品的研制, 国际标准化组织也先后推出一系列建议。高质量高效率的音频压缩技术广泛地用于多媒体应用、音像制品、数字广播、数字电视领域。

#### 4. 音频编码的分类

##### (1) 基于音频数据的统计特性进行编码

其典型技术是波形编码, 其目标是使重建语音波形保持原波形的形状。PCM (脉冲编码调制) 是最简单最基本的编码方法。它直接赋予抽样点一个代码, 没有进行压缩, 因而所需的存储空间较大。为了减少存储空间, 人们寻求压缩编码技术。利用音频抽样的幅度分布规律和相邻样值具有相关性的特点, 提出了差值量化 (DPCM)、自适应量化 (APCM) 和自适应预测编码 (ADPCM) 等算法, 实现了数据的压缩。波形编码适应性强, 音频质量好, 但压缩比不大, 因而数据率较高。

##### (2) 基于音频的声学参数进行参数编码

可进一步降低数据率, 其目标是使重建音频保持原音频的特性。常用的音频参数有共振峰、线性预测系数、滤波器组等。这种编码技术的优点是数据率低, 但还原信号的质量较差, 自然度低。

将上述两种编码算法很好地结合起来, 即采用混合编码的方法, 就能在较低的码率上得到较高的音质。如码本激励线性预测编码 (CELP)、多脉冲激励线性预测编码 (MPLPC) 等。

##### (3) 基于人的听觉特性进行编码

从人的听觉系统出发, 利用掩蔽效应, 设计心理声学模型, 从而实现更高效率的数字音频的压缩。其中以 MPEG 标准中的高频编码和 Dolby AC-3 最有影响。

### 2.3.2 音频编码标准介绍

当前编码技术发展的一个重要方向就是综合现有的编码技术, 制定全球的统一标准, 使信息管理系统具有普遍的互操作性并确保未来的兼容性。国际上, 对语音信号压缩编码的审议在 CCITT 下设的第 15 研究组进行, 相应的建议为 G 系列, 多由 ITU 发表。

国际电报电话咨询委员会 (CCITT) 和国际标准化组织 (ISO) 先后提出一系列有关音频编码的建议, 表 2-2 中列出了一些音频编码算法和国际标准。1972 年首先制定了 G.711 64Kbit/s (A) 律 PCM 编码标准。1984 年又公布了 G.721 标准 (1986 年修订)。它采用的是自适应差分脉冲编码 (ADPCM), 数据率为 32Kbit/s。这两个标准适用于 200~3400Hz 窄带语音信号, 已用于公共电话网。针对宽带语音 (50~7kHz), CCITT 制定了 C.722 编码标准, 它的数据率为 64Kbit/s, 它可用于综合业务数据网 (ISDN) 的 B 通道上传输音频数据。之后公布的 G.723 建议中码率为 40Kbit/s 和 24Kbit/s, G.726 中码率为 16Kbit/s。CCITT 于 1990 年通过了 16~40Kbit/s 镶嵌式 ADPCM 标准 G.727。低码率、短延时、高质量是人们期望的目标。在 AT&T Bell 实验室, 16Kbit/s 短延时码激励 (LD-CELP) 编码方案的基础上, 经优化, CCITT 在 1992 年和 1993 年分别公布了浮点和定点算法的 G.728 标准, 该算法延时小于 2ms, 语音质量可达 MOS4 分以上。ISO 的运动图像专家组在制定运动图像编码标准的同时, 为图像伴音制定了 20KHz 带宽的 128Kbit/s 标准, 1988 年欧洲数字移动通信 GSM 制定了泛美数字移动通信网的 13Kbit/s 长时预测规则码激励 (RPE-LTP) 语音编码标准, 1989 年北美蜂窝电话工业组织 (CTIA) 公布了北美数字移动通信标准, 它采用和时自适应码本激励。日本的数字移动通信标准是 6.7Kbit/s 的 VSELP (矢量和激励线性预测)。

CCITT 正在制定更低码率高质量短延时的音频编码标准。

表 2-2 音频编码算法和标准

	算 法	名 称	资 料 率	标 准	应 用	质 量
波形编码	PCM	均匀量化			公共网 ISDN 配备	4.0~4.5
	$\mu$ (A)	$\mu$ (A)	64Kbit/s	G.711		
	APCM	自适应量化				
	DPCM	差值量化				
	ADPCM	自适应差值量化	32 Kbit/s	G.721		
	SB-ADPCM	子带-自适应 差值量化	64 Kbit/s	G.722		
5.3 Kbit/s 6.3 Kbit/s			G.723			
参数编码	LPC	线性预测编码	2.4 Kbit/s		保密话声	2.5~3.5
混合编码	CELP	码激励 LPC	4.6 Kbit/s		移动通信	4.0~3.7
混合编码	VSELP	向量和激励 LPC	8Kbit/s		语音邮件	
	RPE-LTP	长时预测规则码激励	13.2 Kbit/s		ISDN	
	LD-CELP	低延时码激励 LPC	16 Kbit/s	G.728 G.729		
	MPEG	多子带感知编码	128 Kbit/s		CD	5.0
	AC-3	感知编码			音响	5.0

1. G.711

本建议公布于 1972 年，它给出话音信号编码的推荐特性。话音的取样率为 8kHz，允许偏差是 $\pm 50 \times 10^{-6}$ 。每个样值采用 8 位二进制编码，推荐使用 A 律和 $\mu$ 律编码。本建议中分别给出 A 律和 $\mu$ 律的定义，它是将 13 位的 PCM 按 A 律，14 位 PCM 按 $\mu$ 律转换 8 位编码。简单地讲，建议中把 13（14）PCM 分割成 16 段，各段长度不等，每段给 16 个码字，总编码共 256 个。图 2-9 中示意了 A 律正输入时，输入码与输出码的关系。

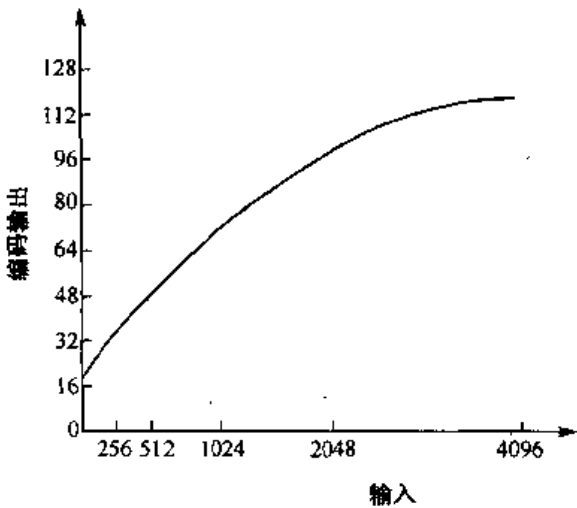


图 2-9 正输入码与 A 律输出码的关系

选用不同译码规律的国家之间，数据通路传送按 A 律译码的信号。使用 $\mu$ 律的国家应进行转换，建议给出了 $\mu$ -A 编码的对应表。建议还规定，在物理介质上连续传输时，符号位在

前，最低有效位在后。

### 2. G.721

这个建议用于 64Kbit/s 的 A 律和  $\mu$  律 PCM 与 32Kbit/s 的 ADPCM 之间的转换。

图 2-10 是 32Kbit/s 的 ADPCM 编码器和解码器的框图。编码器的输入信号是 64Kbit/s 的 A 律和  $\mu$  律 PCM 编码。首先将其转换为标准 PCM 编码，然后从中减去估计值，得到差值信号  $d(k)$ 。15 阶自适应量化器将  $d(k)$  量化成 4 位二进制值  $I(k)$ 。逆量化器从这 4 位二进制数中产生量化的差值信号  $d_q(k)$ 。 $d_q(k)$  和估计值  $S_q(k)$  相加得到重构信号  $S_r(k)$ 。自适应预测器利用  $d_q(k)$  和  $S_r(k)$  生成输入信号的估计值。

解码器包括一个与编码器反馈部分相同的结构，还有 A 律和  $\mu$  律的转换器，以及同步编码调节器。同步编码调节器用于防止同步级联编码（ADPCM-PCM-ADPCM），在某些情况下产生累积失真。用试图消除下一个 ADPCM 编码的量化失真的方式调节 PCM 的输出，以实现同步编码调节。

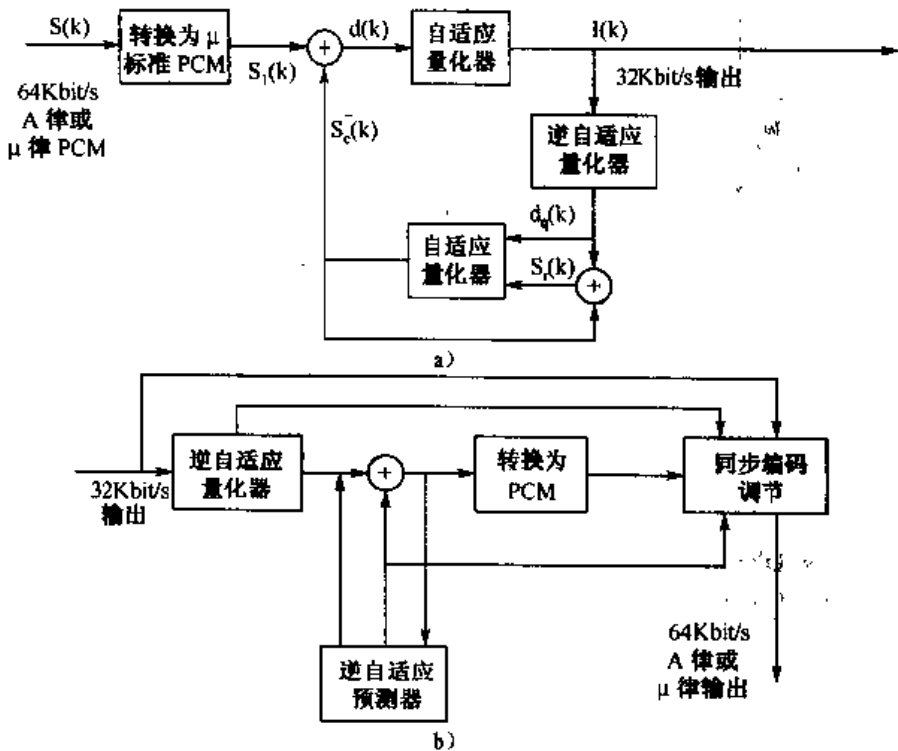


图 2-10 ADPCM 编码器和解码器的框图

a) ADPCM 编码器 b) ADPCM 解码器

### 3. G.722

G.722 建议的带宽音频压缩仍采用波形编码技术，因为要保证既能适用于语音，又能用于其他方式的音频，只能考虑波形编码。G.722 编码采用了高低两个子带内的 ADPCM 方案，高低子带的划分以 4kHz 为界，然后再对每个子带内采用类似 G.721 建议的 ADPCM 编码，因此 G.722 建议的技术方案可以简称为 SB-ADPCM（子带-自适应差分脉冲码调制）。

### 4. G.728

G.728 建议的技术基础是美国 AT&T 公司贝尔实验室提出的 LD-CELP（低延时-码激励线性预测）算法。该算法考虑了听觉特性，其特点是：

- (1) 以块为单位的后向自适应高阶预测。
- (2) 后向自适应型增益量化。
- (3) 以矢量为单位的激励信号量化。

语音输入为每帧 5 个取样值，附加上激励信号的波形与增益表达信息 10 位，编码时延在 2ms 以内。这一点与每一帧取 160 个样值，附加有除激励信号和波形与增益表达信息外还包括线性预测系数、音调预测系数、音调整增益辅助信息等的基本 CELP 结构不同。另外，G.721 方案是对每个取样值进行预测并自适应量化，而本方案则是对所有取样值以矢量为单位处理，并且应用了线性预测和增益自适应的最新理论与成果。编码时将事先准备好的激励矢量的所有组合合成语音，然后将其结果与被编码的输入信号相比较，选出听觉加权后距离最小的码元作为信息传送。而合成器则将发送端编码传送所制定的激励矢量、3 位增益码和自身已合成过的语音波形一起合成为语音。CELP 编码和解码器如图 2-11a、2-11b 所示。

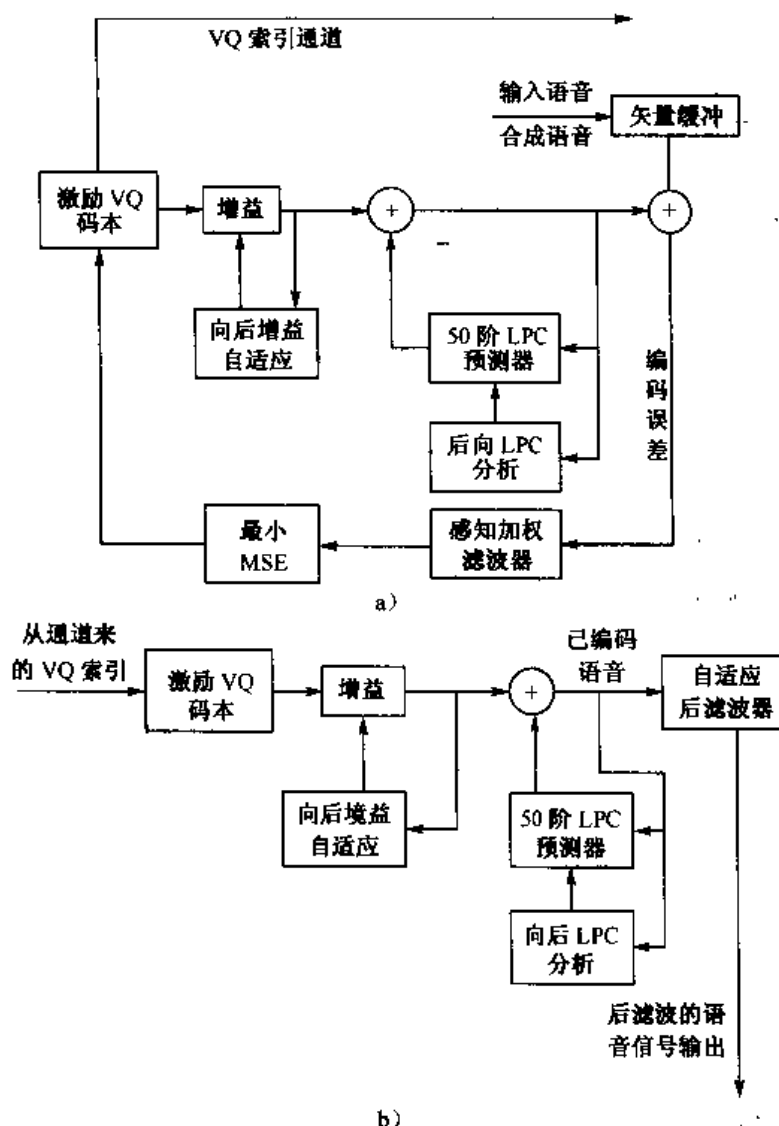


图 2-11 CELP 编码和解码器

a) CELP 编码器 b) CELP 解码器

## 5. MPEG 中的音频编码

国际标准化组织/国际电工委员会所属 WG11 工作组，制定推荐了 MPEG 标准。已公布

和正在讨论的标准有 MPEG-1、MPEG-2、MPEG-4、MPEG-7。本节介绍的内容是 MPEG-1 标准的一部分，对应于 ISO/IEC 11172-3 (MPEG 音频)。这部分规定了高质量音频编码方法、存储表示和解码方法。编码器的输入和解码器的输出与现存的 PCM 标准兼容。ISO/IEC 11172 视频、音频的总数据率为 1.5Mbit/s。音频使用的采样率为 32kHz, 44.1kHz 和 48kHz。编码输出的数据率有许多种，由相关的参数决定。

### (1) 编码器

编码器处理数字音频信号，并生成存储所需的数据流。但编码器的算法并没有标准化，可以使用多种算法，如对音频掩蔽阈值估计的编码、量化和缩放，只要编码器输出的数据能使符合本标准的解码器解出适用的音频流。图 2-12 表明了音频编码器的基本结构。编码过程如下：输入的音频抽样被读入编码器。映射器建立经滤波的输入音频数据流的子带抽样表示。如在层 I、层 II 是子带抽样，在层 III 是经变换的子带抽样。心理声学模型建立一组控制量化和编码的数据，这些数据随实际编码器而变。一种可能的办法是利用音频掩蔽阈值来控制量化器。量化和编码部分是从已映射的输入抽样中生成一组编码符号，这部分也与编码系统有关。帧封装将来自其他模块的输出数据汇集成实际数据，如果需要的话，再加上其他信息，如校正信息。最后输出已编码的数据流。有四种不同的编码模式：单声道模式、双声道模式、立体声模式和联合立体声模式。

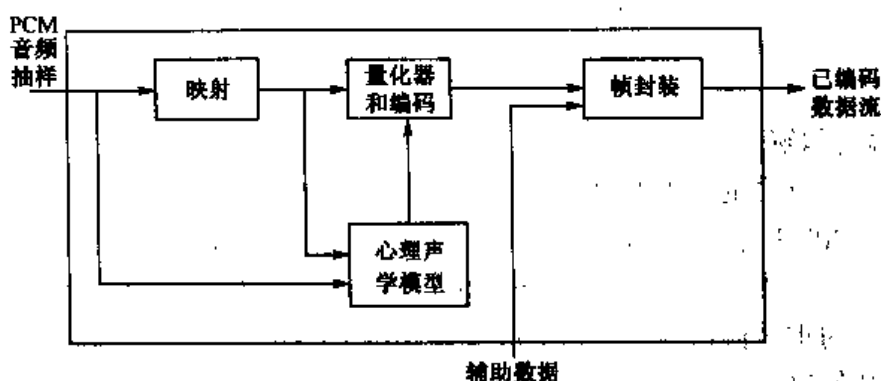


图 2-12 音频编码器基本结构框图

### (2) 编码层次

根据应用需求，可以使用不同层次的编码系统，编码器的复杂性和性能也随之升高。

- 层 I 包括将数字音频变成 32 个子带的基本映射。将数据格式化成块的固定分段。决定自适应位分配的心理声学模型。利用块压扩和格式化的量化器。理论上，层 I 编码/解码的最少延时约为 19ms。
- 层 II 提供了位分配，缩放因子和抽样的附加编码。使用了不同的帧格式。这层理论上的最小编码/解码延时约为 35ms。
- 层 III 采用混合带通滤波器来提高频率分辨率。它增加了差值量化（非均匀）、自适应分段和量化值的熵编码。这层理论上的最小编码/解码延时为 59ms。联合立体声编码作为一个附加特性，能够加入到任何一层中。

### (3) 存储

已编码的视频数据、音频数据、同步数据、系统数据和辅助数据均可一并存入同一存储介质中。如果限定编辑点与可寻地点一致，音频编辑是很容易的。



对存储器的存取可能包括通信系统中的远程存取。假定存取被一个功能单元控制，而不是被音频解码器本身控制。这个控制单元接收用户命令，读取并解释数据的基本结构信息，从介质中读取已存储的信息，分解非音频信息，按所需的速率将存储的音频数据流传送给音频解码器。

#### (4) 解码

解码器按编码器定义的语法接收压缩的音频数据流，按解码部分的方法解出数据元素，按滤波器的规定，用这些信息产生数字音频输出。

图 2-13 表明了音频解码器的基本结构。其解码过程如下：数据流输入到解码器。首先进行数据流拆封，恢复出各种信息。如果在编码器中使用了误差校验，解码器也将进行误差校验。重构单元将重构一组映射抽样的量化方案。逆映射单元把这些抽样变换回均匀 PCM。

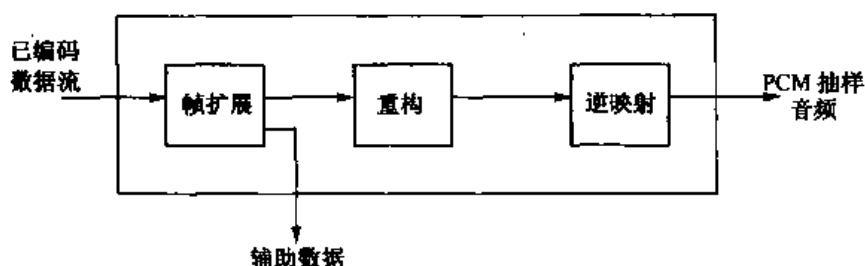


图 2-13 音频解码器结构框图

### 6. AC-3 编码和解码

AC-3 音频编码标准的起源是 DOLBY AC-1。AC-1 应用的编码技术是自适应增量调制 (ADM)，它把 20kHz 的宽带立体声音频信号编码成 512Kbit/s 的数据流。AC-1 曾在卫星电视和调频广播上得到广泛应用。1990 年 DOLBY 实验室推出了立体声编码标准 AC-2，它采用类似 MDCT 的重叠窗口的快速傅立叶变换 (FFT) 编码技术，其数据率在 256Kbit/s 以下。AC-2 被应用在 PC 声卡和综合业务数字网等方面。

1992 年 DOLBY 实验室在 AC-2 的基础上，又开发了 DOLBY AC-3 的数字音频编码技术。AC-3 提供了五个声道的从 20Hz~20kHz 的全通带频响，即正前方的左 (L)、中 (C) 和右 (R)，后边的两个独立的环境声通道左后 (LS) 和右后 (RS)。

AC-3 同时还提供了一个 100Hz 以下的超低音声道供用户选用，以弥补低音不足，因为此声道仅为辅助而已，故定为 0.1 声道，所以 AC-3 被称为 5.1 声道。AC-3 将这 6 个声道进行数字编码，并将它们压缩成一个通道，而它的比特率仅是 320Kbit/s，如图 2-14 所示。

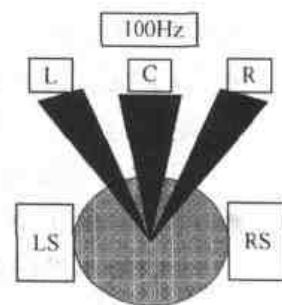


图 2-14 AC-3 5.1 声道

AC-3 除了技术上有许多特点和优点外，在使用上也具有许多优点：它能使单声道的节目内容通过 5 声道得到很好的声音效果；由于在它的比特流内，对每种节目方式，如单声道 (mono)、立体声 (stereo) 和环绕声 (surround) 等都有一个指导信号，因此在工作时，AC-3 能自动地明显地为使用者指示出节目的方式。AC-3 可以将 5.1 通路的信息内容压缩为两个声道供给录制两通道的 VHS 或作为 DOLBY 环绕声的输入源。AC-3 甚至可以将 5.1 通路的信号内容

压缩为单声道，如图 2-15 所示。

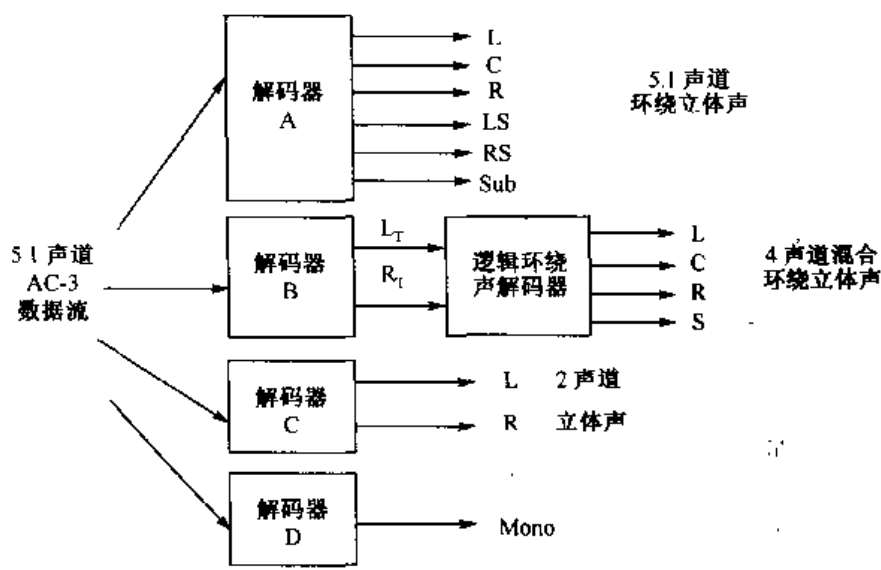


图 2-15 AC-3 可编程解码器

从 1987 年以来，美国一直在考虑日后在美国的高清晰度电视（HDTV）的标准，经过多次的评试和鉴定，DOBLY AC-3 在 1993 年 10 月 25 日被正式定为美国 HDTV 的音频标准，并于 1996 年开始使用。

DOLBY AC-3 的使用范围很广，如制作镭射影碟、CD 唱片、VHS 录像带。在电视广播上，可用在 DBS（数字广播系统）、CATV（有线电视）。此外还应用在 DBS（直播卫星）上。

（1）AC-3 的编码

AC-3 编码器接受标准的 PCM 码流，通过滤波器组变换到频域，然后进行频谱包络分析，根据分析的结果确定相应频率抽样量化所用的存储空间，最后依据 AC-3 语法格式形成码流。编码器框图如图 2-16 所示。

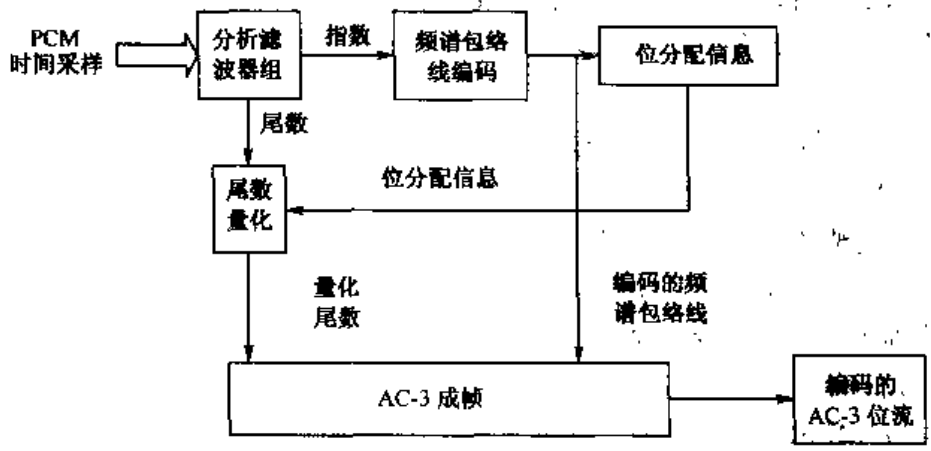


图 2-16 AC-3 编码器框图

下面给出了 AC-3 软件编码器的流程图，如图 2-17 所示，它具体描述了将输入的 PCM 信号编码成 AC-3 数据流的过程。

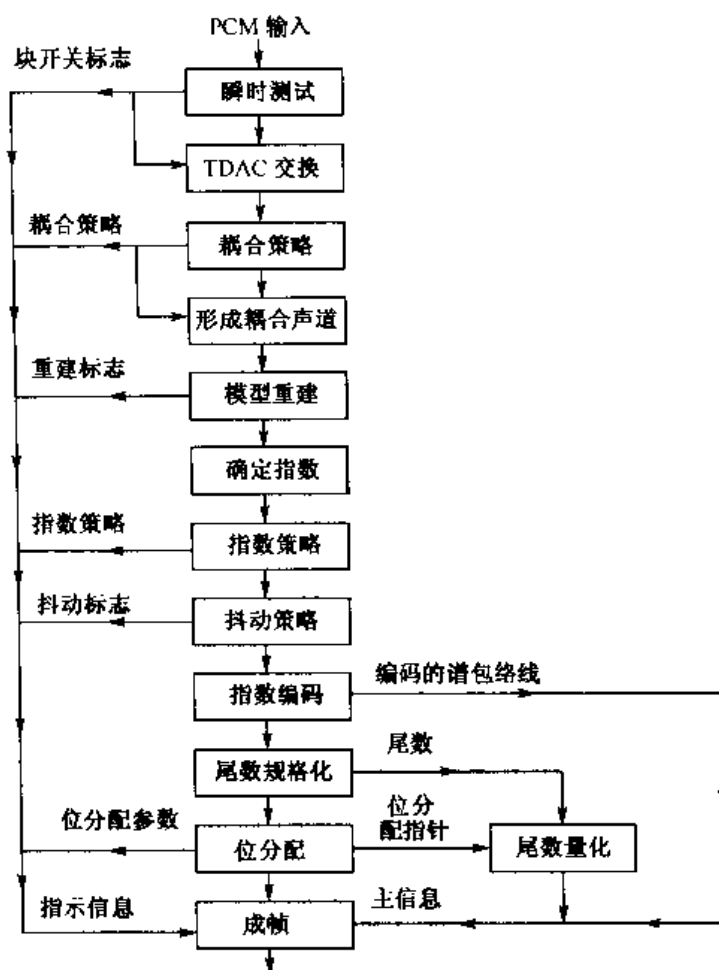


图 2-17 AC-3 编码流程图

AC-3 编码器接收 16 位的 PCM 输入, 经过一些测试后就将时域采样变换成频域采样, 然后根据样点的性质确定是否需要使用耦合, 如果是的话就形成耦合声道。变换系数被分离成指数和尾数来编码, 所谓指数就是变换系数本身的前导零的数目, 尾数是规格化的变换系数。指数用 D15, D25 和 D45 三种方法编码成频谱包, 同时位分配装置产生位分配指针 (bap) 来确定尾数的量化, 然后编码的指数和尾数以及一些同步信息和比特流信息被打包成 AC-3 的一帧输出。

Dolby AC-3 将每一声道的音频根据人耳听觉特性区分为许多最适合的狭窄频段，然后对不同的频段采用不同的编码策略，以达到噪声掩蔽。经此编码处理后，所存的信号在实际放送时仍给人以完整的感受。AC-3 感知编码系统实际上是一种极具选择性的抑噪系统。

Dolby AC-3 还可以用某声道的声压来掩蔽其他声道的噪声，正是这种掩蔽效应使得 Dolby AC-3 达到了很高的数字音频压缩效率，音质也更为逼真。

## (2) AC-3 比特流及语法格式

1) AC-3 与 MPEG 标准类似, 标准只定义压缩数据流的语法和语义, 不为编码器规定固定的结构和实施办法。尤其在听觉模型上每个编码器实施办法是可以完全不同的, 所以其质量可能也是各种各样的。这既有利于应用的规范化和解码器的设计, 又有利于解码算法的改进和发展。

AC-3 比特流由连续的同步帧 (synchronization frame) 组成, 每帧的格式如图 2-18 所示。

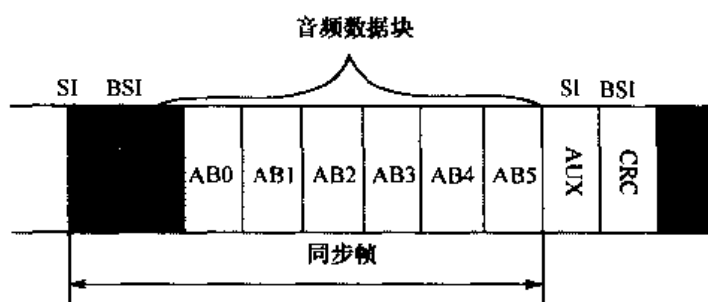


图 2-18 同步帧结构图

同步帧首先是用来保持同步信息（SI），接着是比特流信息头（BSI），之后是六个音频数据块（AB0~AB5），再后是附加数据（AUX）和用于循环冗余校验的 CRC 字。下面对其每一部分进行比较详细的说明。

2) 同步信息。同步信息由以下四部分组成。

- 同步字（syncword）。16 字节长的同步字 0x0B77 作为一帧开始的同步标志。
- 检验码（crc1）。是一个可选的 CRC 校验码，它覆盖一帧的前 5/8。crc2 校验每帧的后 3/8。
- 采样频率（frequency）。说明 PCM 码流的采样频率。
  - 000: 48kHz
  - 001: 44.1kHz
  - 010: 32kHz
  - 011: 保留
- 帧长度代码（frame size code）。与采样率一起指出到下一个同步字的距离。

3) 比特流标识。比特流标识（bsid）中包括了版本、编码模式、LFE 信息、语音平衡、整体压缩等一系列与解码有关的信息。

编码模式包括：

- |                   |                      |
|-------------------|----------------------|
| 000: ch1, ch2     | 001: C               |
| 010: L, R         | 011: L, C, R         |
| 100: L, R, S      | 101: L, C, R, S      |
| 110: L, R, LS, LR | 111: L, C, R, LS, LR |

4) 音频数据块。在音频数据块（audio-block）中包含了音频数据及与之相关的解码控制信息。音频数据块主要记录了频域系数的指数（exponent）和尾数（mantissa）及其控制策略信息，还有动态范围压缩、耦合通道处理、位分配还原附加信息等。根据这些信息，再分配 BSI 和 SI 中的信息，就可以从码流中提取指数，推算出 bit-allocation，再提取尾数，经过反耦合，从而得到各声道的频域系数，再经过反变换，输出时域的 PCM 码，就完成了解码工作。

5) 附加数据。附加数据区保留了用户自己定义的一些信息。

6) 错误校验。包含了一个错误校验字（crc2），它覆盖了整个帧的数据。

### (3) AC-3 解码

由于标准规定了严格的语法格式，所以解码亦成为一种标准的过程。解码器框图如图 2-19 所示。

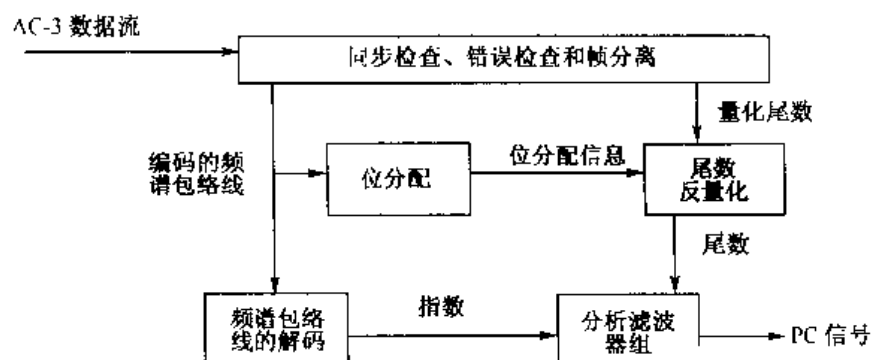


图 2-19 AC-3 解码器框图

AC-3 的解码是与编码不对称的逆过程。它将分析码流的正确性（CRC 校验），然后根据头部信息解出每一声道的指数，继而分析它，并得出相应尾数所占的位数，解出尾数，与指数一起合成频域参数。再通过 IMDCT 和 IFFT 变化，形成标准的 PCM 码流输出。在解码过程中，还要处理多通道耦合、动态范围调整等一系列技术，以达到与压缩前基本相同的质量，同时配合放送装置的控制，达到最佳的再现效果。

## 2.4 小结

声音从模拟信号转换成数字信号仍然占很大的存储空间。为了进一步提高计算机处理音频信号的效率，必须对数字声音信号进行编码处理，在数据通信中已有许多成熟的压缩编码技术和算法，较常用的有脉冲编码调制（PCM）、差分脉冲编码调制（DPCM）和自适应差分编码调制（ADPCM）等其它编码算法和标准等。

目前对音频信号的处理有许多数字信号处理器或者音频卡。典型的是新加坡创新科技有限公司开发的系列产品 Sound Blaster 系列音卡，它的功能有数字音频、音乐合成、MIDI 音效等。它是集语音与音乐于一体的多媒体音频卡，不但具有优良稳定的硬件特性，而且还配有丰富的软件。

随着音频卡的发展和改进，将进一步改善声音质量，统一音频的标准，实现三维环绕立体声、全双工声音处理，与通信技术相结合和单一芯片，另外，简化安装方法——即插即用技术，这些都将使用户使用起来更加简便易行。

## 2.5 习题

1. 数字音频采样和量化过程所用的主要硬件是\_\_\_\_\_。  
 (A) 数字编码器 (B) 数字解码器  
 (C) 模拟到数字的转换器 (A/D 转换器) (D) 数字到模拟的转换器 (D/A 转换器)
2. 音频卡是按\_\_\_\_\_分类的。  
 (A) 采样频率 (B) 声道数 (C) 采样量化位数 (D) 压缩方式
3. 两分钟双声道，16 位采样位数，22.05kHz 采样频率声音的不压缩的数据量是\_\_\_\_\_。  
 (A) 5.05MB (B) 10.58MB (C) 10.35MB (D) 10.09MB

4. 目前音频卡具备以下哪些功能\_\_\_\_\_?
- (1) 录制和回放数字音频文件 (2) 混音  
(3) 语音特征识别 (4) 实时解/压缩数字单频文件  
(A) (1) (3) (4) (B) (1) (2) (4)  
(C) (2) (3) (4) (D) 全部
5. 以下的采样频率中哪个是目前音频卡所支持的\_\_\_\_\_。
- (A) 20kHz (B) 22.05Hz (C) 100kHz (D) 50kHz
6. 1984 年公布的音频编码标准 G.721 采用的是\_\_\_\_\_编码。
- (A) 均匀量化 (B) 自适应量化 (C) 自适应差分脉冲 (D) 线性预测
7. AC-3 数字音频编码提供了五个声道的频率范围, 是\_\_\_\_\_。
- (A) 20Hz~2kHz (B) 100Hz~1kHz  
(C) 20Hz~20kHz (D) 20Hz~200kHz
8. MIDI 的音乐合成器有\_\_\_\_\_。
- (1) FM (2) 波表 (3) 复音 (4) 音轨  
(A) (1) (B) (1) (2) (C) (1) (2) (3) (D) 全部
9. 下列采集的波形声音哪个质量最好\_\_\_\_\_。
- (A) 单声道、8 位量化、22.05kHz 采样频率  
(B) 双声道、8 位量化、44.1kHz 采样频率  
(C) 单声道、16 位量化、22.05kHz 采样频率  
(D) 双声道、16 位量化、44.1kHz 采样频率
10. 简述音频编码的分类及常用编码算法和标准。

## 2.6 参考答案

1. (C)
2. (C)
3. (D)
4. (B)
5. (B)
6. (C)
7. (C)
8. (B)
9. (D)
10. (略)

## 第3章 视频卡和视频信息处理

### 本章要点

- 数字视频信号的获取与处理的基本概念；数字视频的采样、量化的基本原理；彩色空间的表示及转换和彩色全电视信号的组成。
- 视频卡的工作原理、彩色全电视信号的数字锁相和解码器的工作原理，以及视频卡的安装、使用和视频处理软件的使用。

人类感知客观世界有 70% 的信息是由视觉获取的，客观世界的原型应该是景物和图像，语言和文字是对客观世界的一种描述。在日常生活中人们会发现，有时用语言和文字难以表述的事物，用一张简单的图就能够精辟而准确的表达。多少年来图像和视频与计算机一直没有太多的联系，计算机输入输出的信息主要是数字和字符。20 世纪 80 年代末多媒体计算机技术的出现，使计算机具有了综合处理声音、文字、图像和视频信息的能力，它以形象丰富的声、文、图信息和方便的交互性，极大地改善了人机界面，改变了使用计算机的方式，从而为计算机进入人类生活和生产的各个领域打开了大门，它为计算机产业开辟了非常广阔的市场。

### 3.1 彩色空间表示及其转换

多媒体计算机处理图像和视频，首先必须把连续的图像函数  $f(x, y)$  进行空间和幅值的离散化处理，空间连续坐标  $(x, y)$  的离散化，叫做采样； $f(x, y)$  颜色的离散化，称之为量化。两种离散化结合在一起，叫做数字化，离散化的结果称为数字图像。

#### 1. 采样

对连续图像彩色函数  $f(x, y)$ ，沿  $x$  方向以等间隔  $\Delta x$  采样，采样点数为  $N$ ，沿  $y$  方向以等间隔  $\Delta y$  采样，采样点数为  $N$ ，于是得到一个  $N \times N$  的离散样本数组  $[f(m, n)]_{N \times N}$ 。为了达到由离散样本数组以最小失真重建原图的目的，采样密度（间隔  $\Delta x$  与  $\Delta y$ ）必须满足惠特克-卡切尼柯夫-香农（Whittaker-Kotelnikov-shannon）采样定理。

采样定理阐述了采样间隔与  $f(x, y)$  频带之间的依存关系，频带愈窄，相应的采样频率可以降低，采样频率是图像变化频率的二倍时，就能保证由离散图像数据无失真地重建原图。实际情况是空域图像  $f(x, y)$  一般为有限函数，那么它的频域带宽不可能有限，卷积时混叠现象也不可避免，因而用数字图像表示连续图像总会有些失真。

#### 2. 量化

采样是对图像函数  $f(x, y)$  的空间坐标  $(x, y)$  进行离散化处理，而量化是对每个离散点——像素的灰度或颜色样本进行数字化处理。具体说，就是在样本幅值的动态范围内进行分层、取整，以正整数表示。假如一幅黑白灰度图像，在计算机中灰度级以 2 的整数幂表示，即

$G=2^m$ , 当  $m=8, 7, 6, \dots, 1$  时, 其对应的灰度等级为 256, 128, 64,  $\dots, 2$ , 2 级灰度构成二值图像, 画面只有黑白之分, 没有灰度层次, 通常的 A/D 变换设备产生 256 级灰度, 以保证有足够的灰度层次。而彩色幅度如何量化, 要取决于所选用的彩色空间表示。

### 3.1.1 颜色的基本概念

彩色可用亮度、色调和饱和度来描述, 人眼看到任意彩色光都是这三个特性的综合效果。

亮度是光作用于人眼时所引起的明亮程度的感觉, 它与被观察物体的发光强度有关。由于其强度不同, 看起来可能亮一些或暗一些, 显然, 如果彩色光的强度降到使人看不到了, 在亮度标尺上它应与黑色对应, 同样, 如果其强度变得很大, 那么亮度等级应与白色对应。对于同一物体照射的光越强, 反射光也越强, 也称为越亮; 对于不同的物体在相同照射情况下, 反射越强者看起来越亮。此外亮度感还与人类视觉系统的视敏函数有关, 即便强度相同, 不同颜色的光当照射同一物体时也会产生不同的亮度。

色调是当人眼看一种或多种波长的光时所产生的彩色感觉, 它反映颜色的种类, 是决定颜色的基本特性。红色、棕色等都是指色调。某一物体的色调, 是指该物体在日光照射下, 所反射的各光谱成分作用于人眼的综合效果, 对于透射物体则是透过该物体的光谱综合作用的结果。

饱和度是指颜色的纯度即掺入白光的程度, 或者说是指颜色的深浅程度, 对于同一色调的彩色光, 饱和度越深颜色越鲜明或叫做越纯。例如, 当红色加进白光之后冲淡为粉红色, 其基本色调还是红色, 但饱和度降低, 换句话说, 淡色的饱和度比鲜色要低一些。饱和度还和亮度有关, 因为若在饱和的彩色光中增加白光的成分, 增加了光能, 因而变得更亮了, 但是它的饱和度却降低了。如果在某色调的彩色光中, 掺入别的彩色光, 则会引起色调的变化, 只有掺入白光时才引起饱和度的变化。

通常把色调和饱和度通称为色度, 上述内容总结为: 亮度表示某彩色光的明亮程度, 而色度则表示颜色的类别与深浅程度。

三基色 (RGB) 原理: 自然界常见的各种彩色光, 都可由红 (R)、绿 (G)、蓝 (B) 三种颜色光按不同比例相配而成, 同样绝大多数颜色也可以分解成红、绿、蓝三种色光, 这就是色度学中最基本的原理——三基色原理。当然三基色的选择不是唯一的, 也可以选择其它三种颜色为三基色, 但是, 三种颜色必须是相互独立的, 即任何一种颜色都不能由其它两种颜色合成。由于人眼对红、绿、蓝三种色光最敏感, 因此由这三种颜色相配所得的彩色范围也最广, 所以一般都选这三种颜色作为基色。

把三种基色光按不同比例相加称之为相加混色, 由红、绿、蓝三基色进行相加混色的情况如下:

红色+绿色=黄色

红色+蓝色=品红

绿色+蓝色=青色

红色+绿色+蓝色=白色

称黄、品红和青色为相加二次色, 此外还可以看出:

红色+青色=绿色+品红=蓝色+黄色=白色

我们称青色、品红和黄色分别是红、绿、蓝三色的补色。



由于人眼对于相同亮度单色光的主观亮度感觉不同，所以，用相同亮度的三基色混色时，如果把混色后所得单色光亮度定为 100% 的话，那么人的主观感觉是绿光仅次于白光，是三基色中最亮的；红光次之，亮度约占绿光的一半；蓝光最弱，亮度约占红光的 1/3。当白光的亮度用  $Y$  来表示时，它和红、绿、蓝三色的关系可用如下的方程描述：

$$Y=0.299R+0.587G+0.114B \tag{3-1}$$

这就是常用的亮度公式，它是根据美国国家电视制式委员会的 NTSC 制式推导得到的，如果采用 PAL 电视制式时，白光的亮度公式将作如下改动：

$$Y=0.222R+0.707G+0.071B \tag{3-2}$$

两个公式不同的原因，是由于所选取的显示三基色不同，三基色及其补色的亮度比例图如图 3-1 所示，其中三补色亮度比例等于合成补色的基色亮度比例之和。

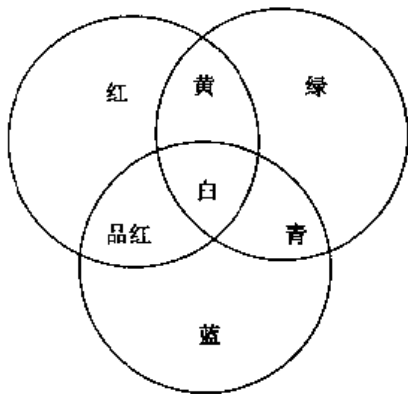


图 3-1 相加混色之三基色及其补色

### 3.1.2 彩色空间表示

#### 1. RGB 彩色空间

在多媒体计算机技术中，用的最多的是 RGB 彩色空间表示，因为计算机彩色监视器的输入需要 RGB 三个彩色分量，通过三个分量的不同比例，在显示屏幕上合成所需要的任意颜色，所以不管在多媒体系统中采用什么形式的彩色空间表示，最后的输出一定要转换成 RGB 彩色空间表示。

在 RGB 彩色空间，任意彩色光  $F$ ，其配色方程可写成：

$$F=r[R]+g[G]+b[B] \tag{3-3}$$

其中  $r$ 、 $g$ 、 $b$  为三系数， $r[R]$ 、 $g[G]$ 、 $b[B]$  为  $F$  色光的三色分量。任意一种色光，其色度可由相对色系数中的任意两个惟一确定。因此，各种彩色的色度可以用二维函数表示。用  $r$  和  $g$  作为直角坐标系中两个直角坐标所画的各种色度的平面图形，就叫 RGB 色度图，如图 3-2 所示。

#### 2. YUV 和 YIQ 彩色空间

在现代彩色电视系统中，通常采用三管彩色摄像机或彩色 CCD 摄像机，它把摄得的彩色图像信号经分光棱镜分成  $R_0G_0B_0$  三个分量的信号，分别经放大和  $\gamma$  校正得到 RGB，再经过矩阵变换电路得到亮度信号  $Y$ 、色差信号  $R-Y$  和  $B-Y$ ，最后发送端将  $Y$ 、 $R-Y$  及  $B-Y$  三个信号进行编码，用同一信道发送出去。这就是我们常用的 YUV 彩色空间，采用 YUV 彩色

空间的好处如下：

(1) 亮度信号 Y 解决了彩色电视机与黑白电视机的兼容问题。

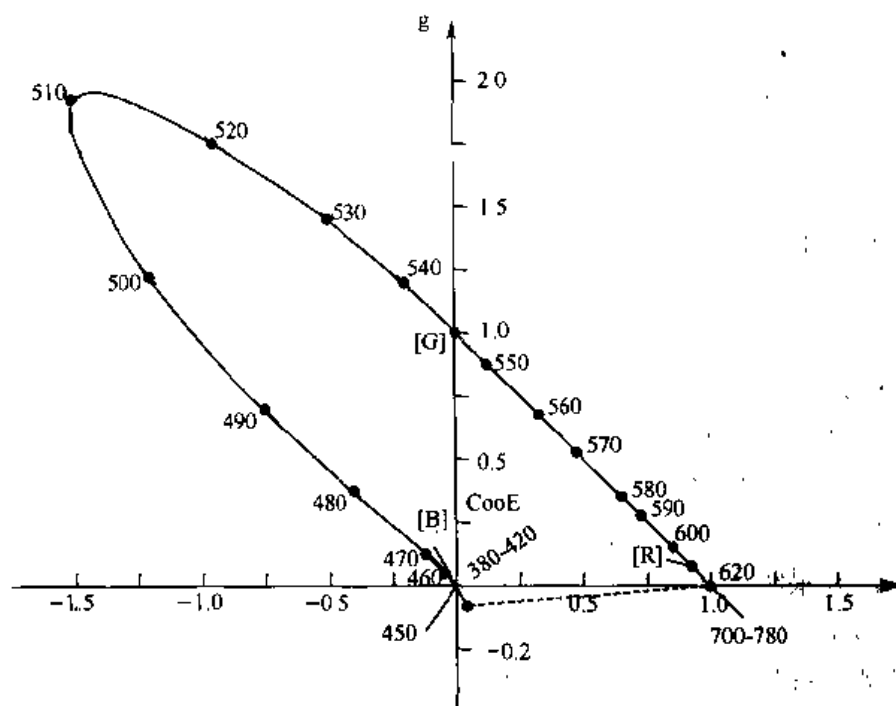


图 3-2 RGB 色度图

(2) 大量实验表明，人眼对彩色图像细节的分辨本领比对黑白的低得多，因此对色度信号 U、V，可以采用“大面积着色原理”。用亮度信号 Y 传送细节，用色差信号 UV 进行大面积涂色，因此彩色图像的清晰度由亮度信号的带宽保证（PAL 制亮度信号 Y 的带宽采用 4.43MHz），而把色度信号的带宽变窄（PAL 制色度信号带宽限制在 1.3MHz）。

正是由于这个原因，在多媒体计算机中采用了 YUV 彩色空间，数字化后通常为 Y:U:V = 8:4:4 或者是 Y:U:V = 8:2:2，后者具体的作法是把亮度信号 Y 的每个像素都数字化为 8b（256 级亮度），而 U、V 色差信号每四个像素用一个 8b 数据表示，即粒度变大。将一个像素用 24b 表示压缩为用 12b 表示，而人的眼睛却感觉不出来。

美国、日本等国采用的 NTSC 制，选用了 YIQ 彩色空间，Y 仍为亮度信号，I、Q 仍为色差信号，但它们与 U、V 是不同的，其区别是色度向量图中的位置不同，如图 3-3 所示，Q、I 为互相正交的坐标轴，它与 U、V 正交轴之间有 33° 夹角。

由图 3-3 可知 I、Q 与 V、U 之间的关系可以表示成：

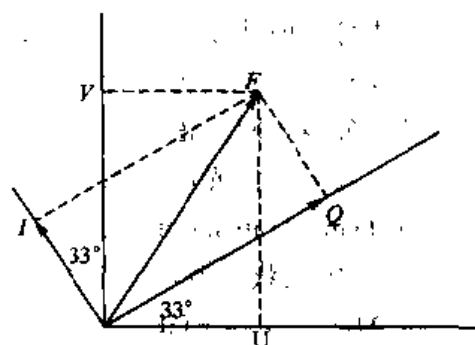


图 3-3 IQ 轴与 UV 轴的关系

$$\begin{cases} I = V \cos 33^\circ - U \sin 33^\circ \\ Q = V \sin 33^\circ + U \cos 33^\circ \end{cases} \quad (3-4)$$

选择 YIQ 彩色空间的好处是，人眼的彩色视觉特性表明人眼分辨红、黄之间颜色变化

的能力最强，而分辨蓝与紫之间颜色变化的能力最弱。在色度向量图中，人眼对于处在红、黄之间，相角为  $123^\circ$  的橙色及其相反方向相角为  $303^\circ$  的青色，具有最大的彩色分辨力，因此把通过  $123^\circ$  至  $303^\circ$  线即 IO 线的色度信号称为 I 轴，它表示人眼最敏感的色轴。与 I 正交的色度信号轴称为 Q 轴，表示人眼最不敏感的色轴。在传送分辨力弱的 Q 信号时，可用较窄的频带，而传送分辨力较强的 I 信号时，可用较宽的频带。在 NTSC 制中，I 的带宽取  $1.3\sim 1.5\text{MHz}$  和 PAL 制的 U、V 带宽差不多，而 Q 的传送带宽只是  $0.5\text{MHz}$ ，仅是 I 带宽的  $1/3$ 。

### 3. HSI 彩色空间

在 HSI 彩色空间中，人们常用 H、S、I 三参数描述颜色特性，其中 H 表示色调 (Hue)，S 表示颜色的饱和度 (Saturation)，I 表示光的强度 (Intensity)。用一个三维空间的枣形立体图 (见图 3-4) 表示可以把颜色的这三个参量的特性更加形象的表示出来。

在颜色立体图中，垂直轴表示光的亮度变化，顶部最亮表示白色，底最暗表示黑色，中间是介于白黑之间深浅不同的灰度。与黑白轴垂直的水平面圆周上，各点代表光谱上各种不同的色调，如图中箭头所指 (红、橙、黄……紫、红) 构成闭合的圆环。处于圆周上的点是饱和的颜色。圆周上各点与圆形中心点的亮度相同，从圆周到圆心过渡表示颜色饱和度的逐渐降低，当颜色在枣形立体图同一平面上变化时，只改变色调和饱和度而亮度不变。

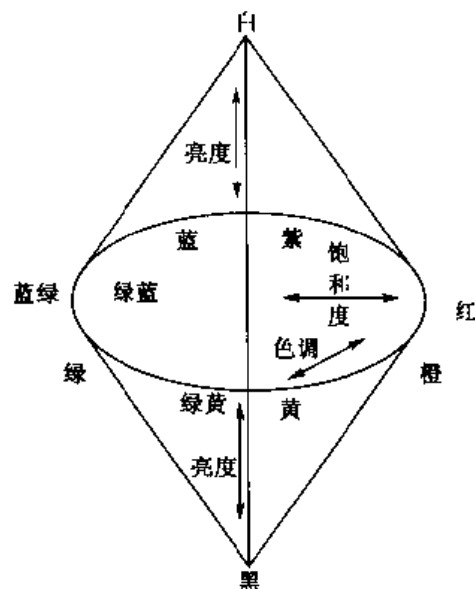


图 3-4 颜色立体图

枣形立体图是一个理想化的模型，在真实的颜色关系中，饱和度最大的黄色并不在中等亮度的地方，而是在靠近白色较高的地方；饱和度最大的蓝色则靠近黑色亮度较低的地方。因此，颜色立体中部的色调图形平面是一个斜平面，黄色部分较高，蓝色部分较低。就人眼的彩色视觉特性而言，用色调、饱和度、光强 (亮度) 描述彩色光是合适的。色调决定彩色光的光谱成分，取决于光的波长，说明彩色光中混入白光的数量。饱和度是某种波长的彩色光纯度的反映，纯光谱色的含量越多，其饱和度越高，高饱和度的彩色光颜色深，当光谱色掺入白光成分越多时，饱和度下降，颜色变浅。亮度决定于彩色的强度，是彩色光对视觉的刺激程度，表征彩色光所含的能量特征，能量大显得亮，反之，则显得暗。

美国 Data Translation 公司生产的视频信号获取器 DT2871HSI (Hue, Saturation and Intensity) 彩色帧获取器，就采用 HSI 彩色空间，原理方框图如图 3-5 所示。RGB 输入后，经过三路 A/D 变换和 RGB/HSI 转换，将 HSI 数字信号存在  $512 \times 512 \times 8 \times 3$  ( $256\text{KB} \times 3$ ) 帧内存中，在显示输出之前还要经过 HSI/RGB 彩色空间转换以及三路 D/A 变换，得到模拟的 RGB 信号送给彩色监视器显示。

采用 HSI 彩色空间能够减少彩色图像处理的复杂性，而增加快速性，它更接近人对彩色的认识 and 解释。例如色调 (Hue) 是颜色的属性，它描述真正的彩色，如纯红、纯黄、纯绿、纯蓝、纯紫以及它们之间的某些颜色。当我们说颜色这个词时色调是最能直接说明彩色这个概念。饱和度 (Saturation) 是颜色另一个属性，它描述纯颜色用白色冲淡的程度，高

饱和度的颜色含有较少的白色。亮度是非彩色属性，它描述亮还是暗。彩色图像中的亮度对应于黑白图像中的灰度。在图像处理中常用的算术操作或算法，例如作为边缘检测或边缘增强的 Sobel 操作数（卷积运算），只要对 HSI 彩色空间的亮度信号进行操作就可获得良好效果，而在 RGB 彩色空间要作上述运算就很不方便。在图像处理和计算机视觉中大量算法都可在 HSI 彩色空间中方便地使用，它们可以分开处理而且是相互独立的。因此，在 HSI 彩色空间中可以大大简化图像分析和处理的工作量。

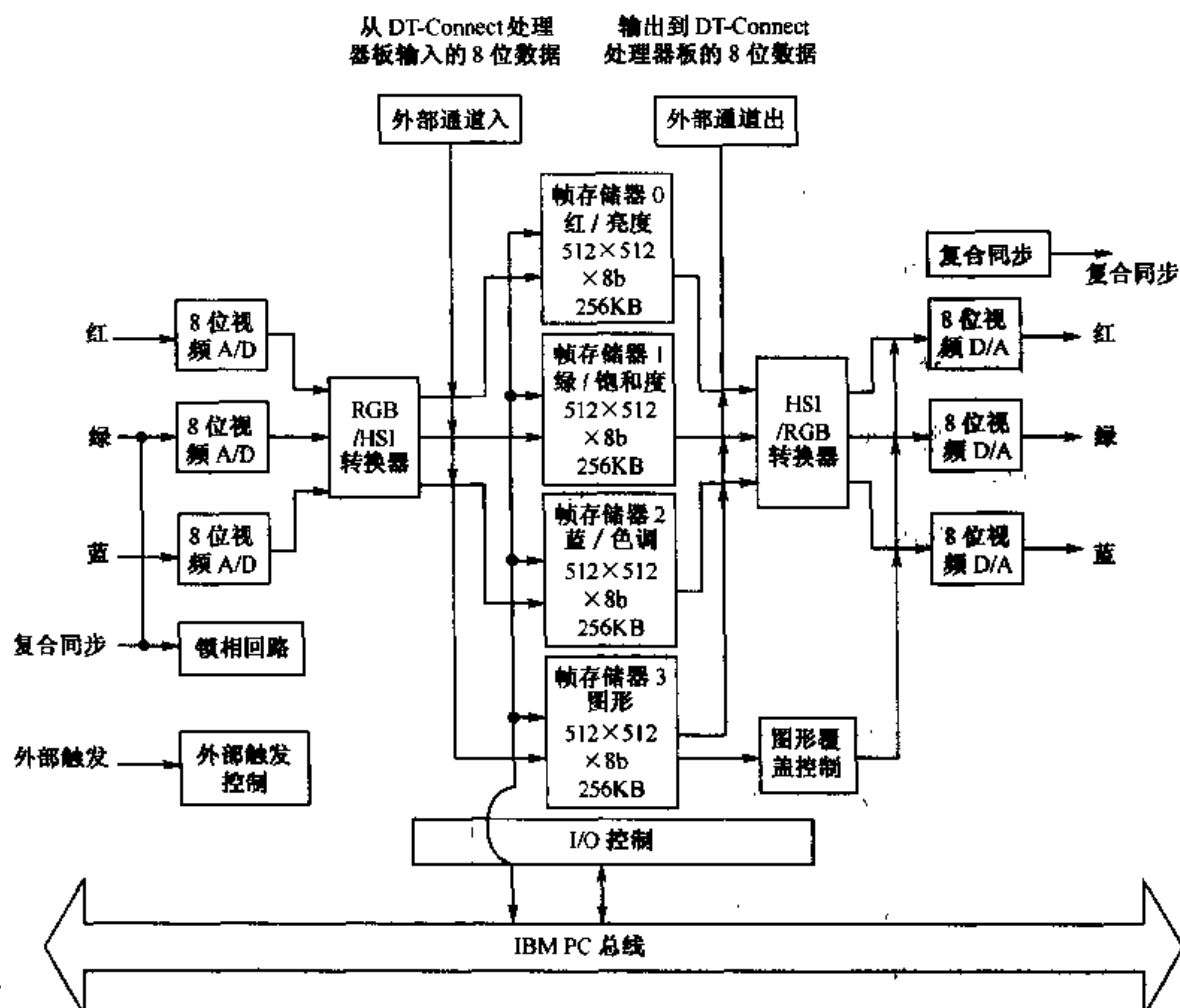


图 3-5 DT2781 (HSI) 彩色帧获取器原理方框图

#### 4. 其他彩色空间表示

彩色空间表示还有很多种，如 CIE（国际照明委员会）制定的 CIE XYZ, CIE LAB 彩色空间，CCIR（Consultative Committee International Radio）制定的 CCIR601-2YCbCr 彩色空间。

按照 CCIR601-2 建议，欧洲电视专家组将非线性的 RGB 信号编码成 YCbCr，编码过程开始是先采用符合 SMPTE-CRGB（它定义了三种荧光粉及一种参考白光，应用于演播室监视器及电视接收机标准的 RGB）基色作为  $\gamma$  校正信号。非线性 RGB 信号很容易与一个常量矩阵相乘而得到亮度 Y 和两个色差信号 Cb 和 Cr。靠近中心轴的彩色，其亮度信号与 CIE XYZ 的 Y 成分的  $\gamma$  校正形式非常接近，CCIR601-2YCbCr 通常在图像压缩时选作彩色空间，而在通信中是一种非正式标准。

### 3.1.3 彩色空间的转换及其实现技术

#### 1. RGB 与 YUV 和 YIQ 之间的转换

彩色摄像机最初得到的是经过 $\gamma$ 校正的 RGB 信号，为了和黑白电视机兼容及压缩编码，在传送过程中包含亮度信号和色差信号，亮度方程简化如下：

$$Y = 0.3R + 0.59G + 0.11B \quad (3-5)$$

公式表明，用三基色显示彩色时，各基色组成亮度  $Y$  的比例关系是恒定的。这些比例系数有时称之为“可见度系数”，它们的和为 1，这表示当基色信号电压  $E_R$ 、 $E_G$ 、 $E_B$  各为 1V 时，构成亮度信号  $E_Y$  也是 1V。

三个色差信号  $B-Y$ 、 $R-Y$ 、 $G-Y$  中有两个是独立的，最后一个可用亮度方程和两个色差信号通过运算得到，表达式如下：

$$\begin{cases} B-Y = B - 0.3R - 0.59G - 0.11B = -0.3R - 0.59G + 0.89B \\ R-Y = R - 0.3R - 0.59G - 0.11B = 0.7R - 0.59G - 0.11B \\ Y = 0.3R + 0.59G + 0.11B \end{cases} \quad (3-6)$$

为了达到彩色与黑白兼容，要求传输的动态范围满足亮度信号的要求，如果按上述方法传输彩色全电视信号，会造成幅度失真，为此必须对彩色信号进行压缩，压缩方法是让色差信号乘上一个小于 1 的压缩系数：

$$\begin{cases} U = m(B-Y) = 0.493(B-Y) \\ V = n(R-Y) = 0.877(R-Y) \end{cases} \quad (3-7)$$

将式 (3-6) 代入式 (3-7)，整理后得到：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.3 & 0.59 & 0.11 \\ -0.15 & -0.29 & 0.44 \\ 0.61 & -0.52 & -0.096 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3-8)$$

YIQ 彩色空间和 RGB 彩色空间的转换方法是：将  $V=0.877(R-Y)$ ， $U=0.493(B-Y)$ ， $\sin 33^\circ=0.545$ ， $\cos 33^\circ=0.839$  代入式 (3-4)，可得到：

$$\begin{cases} I = 0.74(R-Y) - 0.27(B-Y) \\ Q = 0.48(R-Y) + 0.41(B-Y) \end{cases} \quad (3-9)$$

将式 (3-6) 代入式 (3-9)，整理后得到：

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.3 & 0.59 & 0.11 \\ 0.6 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3-10)$$

#### 2. HSI 彩色空间与 RGB 彩色空间的转换

HSI 彩色空间为多媒体计算机和计算机视觉彩色图像实时处理和特技提供了有效方法，HSI 彩色空间的三个帧内存 (Frame Buffer) 的资料在处理彩色图像时相互是独立的，分别提供解释彩色图像非常有用的信息。一幅彩色图像很容易从 RGB 彩色空间转换到 HSI 彩色

空间，具体公式如下：

$$I = \frac{R+G+B}{3} \quad (3-11)$$

$$H = \frac{1}{360} \left[ 90 - \arctan(F/\sqrt{3}) + \{0, G > B; 180, G < B\} \right] \quad (3-12)$$

其中，

$$F = \frac{2R-G-B}{G-B} \quad (3-13)$$

$$S = 1 - \left[ \frac{\min(R, G, B)}{I} \right] \quad (3-14)$$

如果 RGB 的色度轴可以想像是由摄像机产生的 RGB，该颜色的亮度就由式 (3-11) 确定。色调信号用色平面上的角度来表示。具体方法是：把白色点 ( $R=G=B$ ) 与红色顶点的联机作为基准，即该线的角度为零。色度就是色度平面上任意点与白色点的联机和此基准线的夹角，具体算法可由式 (3-12) 和 (3-13) 给出。饱和度  $S$  是从 1 中减去  $R/I$ 、 $G/I$ 、 $B/I$  的最小值，白色是  $R/I=G/I=B/I=1$ ，所以它的饱和度值为零。

图 3-6 给出一个处理的例子，在 RGB 彩色空间，如果一个像素的颜色从黄色变成蓝色，对应的三个帧内存的值变化如下：143→31（红）、143→31（绿），31→255（蓝）。而在 HSI 彩色空间，同样的变化，对应的三个帧内存的值，只要色调  $H$  从 60 变化到 240，其它饱和度和亮度都不要变化。从这个例子可以清楚地看到，在处理彩色问题时，HSI 彩色空间具有明显的优越性。

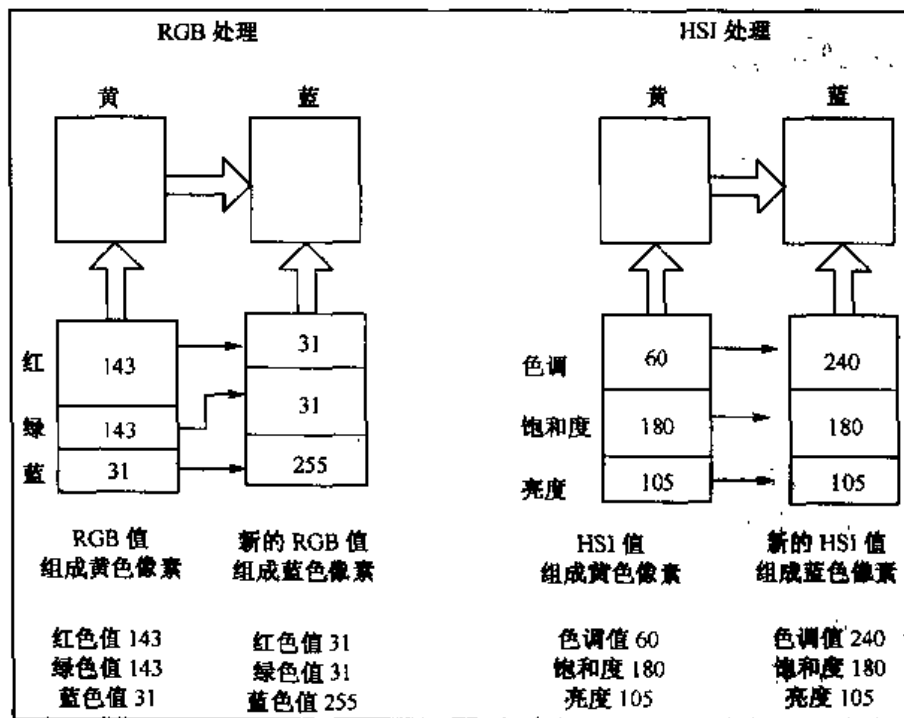


图 3-6 RGB 彩色空间和 HSI 彩色空间处理彩色问题的比较

多媒体计算机系统涉及到多种彩色空间，在 CD-I 系统中，支持的图像格式、视频方式有 DYUV、RGB5:5:5、CLUT8、CLUT7、CLUT4、RL7 以及 RL3。其中 DYUV 方式与

CCIR601-2YCbCr 有密切联系；RGB5:5:5 方式采用 RGB 彩色空间，RGB 每个分量占用 5 位；CLUT8、CLUT7、CLUT4、RL7 以及 RL3 采用的也是 RGB 彩色空间，它用彩色查找表（Color Look-up Table-CLUT，也叫调色板）进行映像和译码。在 DVI 系统中采用 YUV 彩色空间，它所支持的图像格式又涉及到 YIQ、RGB 彩色空间，其中 RGB 支持 8 位、16 位和 24 位。综上所述，搞清楚彩色空间表示以及它们之间的转换，是多媒体计算机彩色图形、静态图像以及动态图像（Video）处理算法的基础。

### 3.1.4 全电视信号

电视摄像机是一种广泛使用的视频和图像的输入设备，它能将景物、图片等光学信号转变为全电视信号，目前主要有黑白和彩色两种摄像机。

#### 1. 黑白全电视信号

电视摄像机把一幅图像信号转变成的输出信号就是全电视信号。全电视信号主要由图像信号（视频信号）、复合消隐信号和复合同步信号组成。这两种信号加在一起称为全电视信号，其波形如图 3-7 所示。

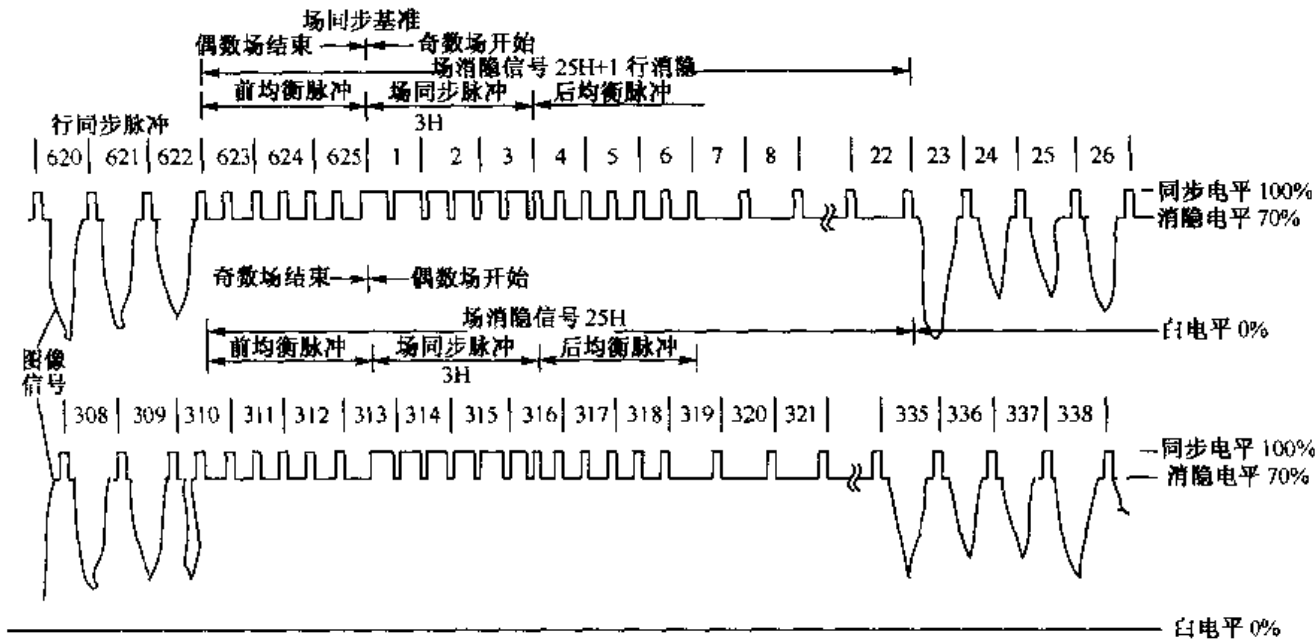


图 3-7 全电视信号

全电视信号的幅度是：以同步信号作为 100%，黑电平和消隐电平为 70%，白电平为 0%，图像信号介于白电平和黑电平之间，根据图像的灰度而变化。在标准的 1V 全电视信号中，同步信号为  $0.3V \pm 9mV$ ，图像信号为  $0.7V \pm 20mV$ 。

从时间上看，每一行的周期为  $64\mu s$ ，其中，图像占  $52.2\mu s$ ，行消隐占  $11.8\mu s \pm 250ns$ 。行同步信号的带宽为  $4.7\mu s \pm 100ns$ ，它比行消隐信号延迟  $1.3\mu s \pm 250ns$ 。每一场的周期为  $312.5H = 20ms$ ，其中，场消隐信号占  $25H + 1$  行消隐信号，即等于  $1600\mu s + 11.8\mu s$ 。均衡脉冲的宽度是  $2.35\mu s \pm 100ns$ ，周期为半行，共 12 个（前六个，后六个）。场同步脉冲有 6 个槽脉冲，其宽度为  $4.7\mu s \pm 100ns$ 。

在全电视信号中，把奇数场同步信号的前沿作为一场的起点，第 1、2、3 行是场同步信

号，第 4、5、6 行是后均衡脉冲，7~22 行还是场消隐信号，该场消隐信号从前场 623 行开始，因此，整个消隐信号是 25 行加一个行消隐时间。图像信号从 23 行起到 309.5 行止，共 287.5 行，这就是第一场或称奇数场。从 309.5 行开始又是下一场的场消隐信号及前均衡脉冲，在 312.5 行出现下一个偶数场的场同步脉冲，奇数场到此结束。偶数场开始，图像信号及偶数场结束，如图 3-7 所示。奇数场加上偶数场称为一帧。

### 2. 彩色全电视信号

在现代彩色电视系统中，通常采用 YUV 彩色空间或 YIQ 彩色空间，Y 为亮度信号，它可以与黑白全电视信号兼容，U 和 V 用载波频率 $\omega_{sc}$ 调制加到亮度 Y 上，最后形成彩色全电视信号，如下式所示：

$$C_{vbs} = Y + U\cos\omega_{sc}t + S(t)\sin\omega_{sc}t \tag{3-15}$$

在 NTSC 制系统中，U 信号调制在副载波的零相位上，而 V 信号是固定地调制在 90° 的相位上的。在 PAL 制系统中，调制情况略有差别。U 信号的调制与 NTSC 制相同，而 V 信号的调制是：第一行调制在 90° 的相位上（与 NTSC 制相同，称为 NTSC 行）；下一行（同隔行扫描是下面的第三行）调制在 270° 的相位上（称为 PAL 行）；再下一行又回到 90° 的相位上。按此顺序，V 信号调制相位逐行倒相 180°。图 3-8 所示为 PAL 制的平衡正交调制的倒相原理。

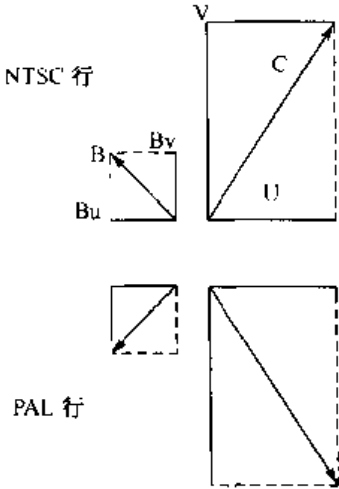


图 3-8 PAL 制平衡正交调制中的倒相原理

根据图 3-8 的向量图，可写出 PAL 制色度信号 Ch 的表达式：

$$Ch = U\cos\omega_{sc}t + S(t)\sin\omega_{sc}t \tag{3-16}$$

式中，S(t)称为 PAL 开关函数，它是双极性矩形脉冲，其重复周期为行周期 Th 的两倍，幅度为+1 和-1。PAL 开关函数 S(t)代表 PAL 制系统的根本特征，它的引入相当有效地克服了 NTSC 制系统对信道微分相位敏感的缺点，这是 PAL 制取得成功的原因。

最后将亮度、复合消隐信号与色度信号、复合同步信号混合放大，形成 PAL 制彩色全电视信号。以 100% 幅度和 100 % 饱和度（简称为 100/100）的彩条信号为例的彩色全电视信号如图 3-9 所示，图中标出了各部分的标准电平数值。

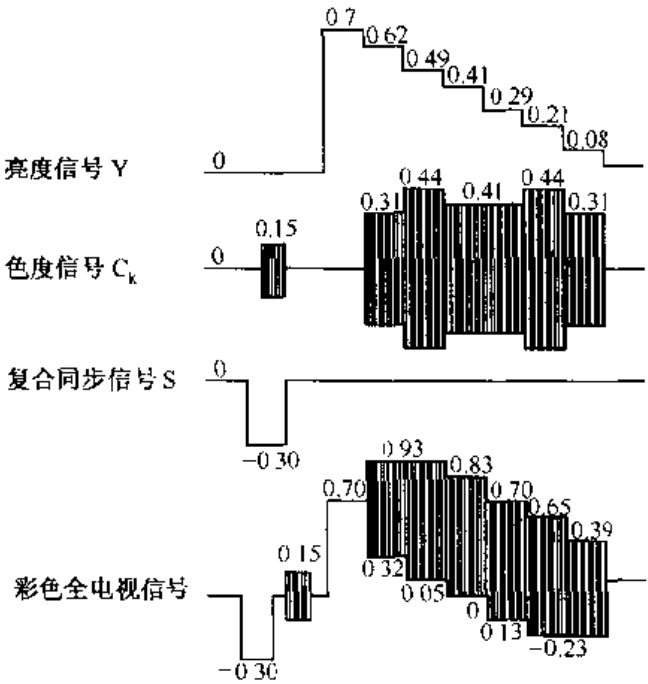


图 3-9 在 75Ω负载上，100/100 彩条的彩色全电视信号各部分的电平标准（输入信号 R、G、B 幅度为 0.7V 时）



送到 D/A 变换器中变成模拟的 RGB 图像，同时送到数字式视频编辑器中进行视频编码，最后输出到 VGA 监视器及电视机或录像机。根据图 3-10 我们将它分成六大部分，简述如下：

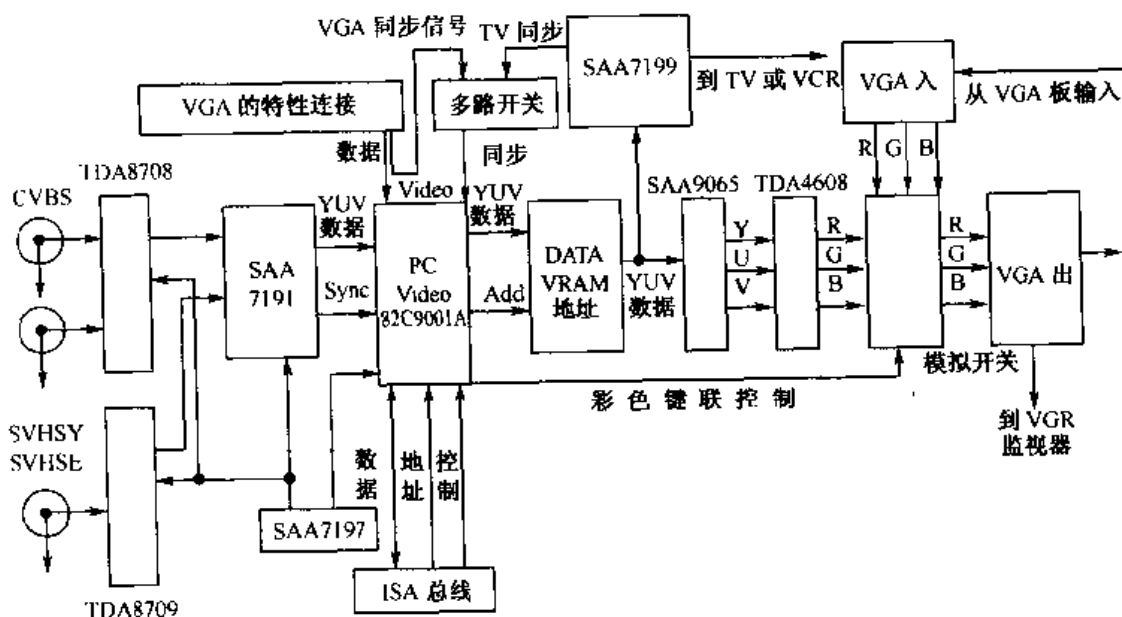


图 3-10 多媒体计算机视频信号获取器总体框图

### 1. A/D 变换和数字译码

从彩色摄像机、录像机或其他视频信号源得到的彩色全电视信号，首先送到视频模拟输入埠，即 TDA8708 或 TDA8709 进行 A/D 变换，TDA8708 的工作原理如图 3-11 所示：它有三个视频输入端通过编程可控制视频选择位 0 和选择位 1，选中 3 个输入端的任意一个作为输入，然后送到具有钳位电路和自动增益功能的运算放大器中，最后经过 A/D 变换器将彩色全电视信号转换成 8 位数字信号，送给彩色多制式数字译码器。A/D 变换器的时钟、同步脉冲以及黑电平的同步脉冲由多制式数字译码电路提供。

全电视信号的峰值电平比较器、黑电平比较器及同步电平比较器与运算放大器的自动增益控制 (AGC) 电路一起, 减少了放大器的零点漂移, 保证了 A/D 变换器在线性范围内工作。

SAA7191 是数字式多制式译码器, 它接受 8 位 CVBS 或 8 位 Y、8 位 C super VHS 输入, 支持 PAL-B/G、NTSC-M、SECAM 等多种电视彩色编码制式。在只使用一个 26.8MHz 的晶体管时, 它所支持的水平有效像素为 768 (50Hz 场频) 或 640 (60Hz 场频), 它的工作状态可通过 I<sup>2</sup>C 总线进行控制。

经SAA7191译码后输出的信号有两种格式,即 $Y:U:V=4:1:1$ 和 $Y:U:V=4:2:2$ ,其资料组织格式如表3-1所示。由此表可知,4:1:1格式是4个Y像素共享一对U、V,4:2:2格式是2个Y像素共享一对U、V。如前所述,人眼对亮度信号分辨敏感,对彩色空间信号分辨不敏感,因此不必使U、V彩色空间分辨率与Y亮度分辨率相同。事实上 $Y:U:V=4:1:1$ 的图形质量已达到了一般的电视要求,而 $Y:U:V=4:2:2$ 的图像质量已是广播级的了。

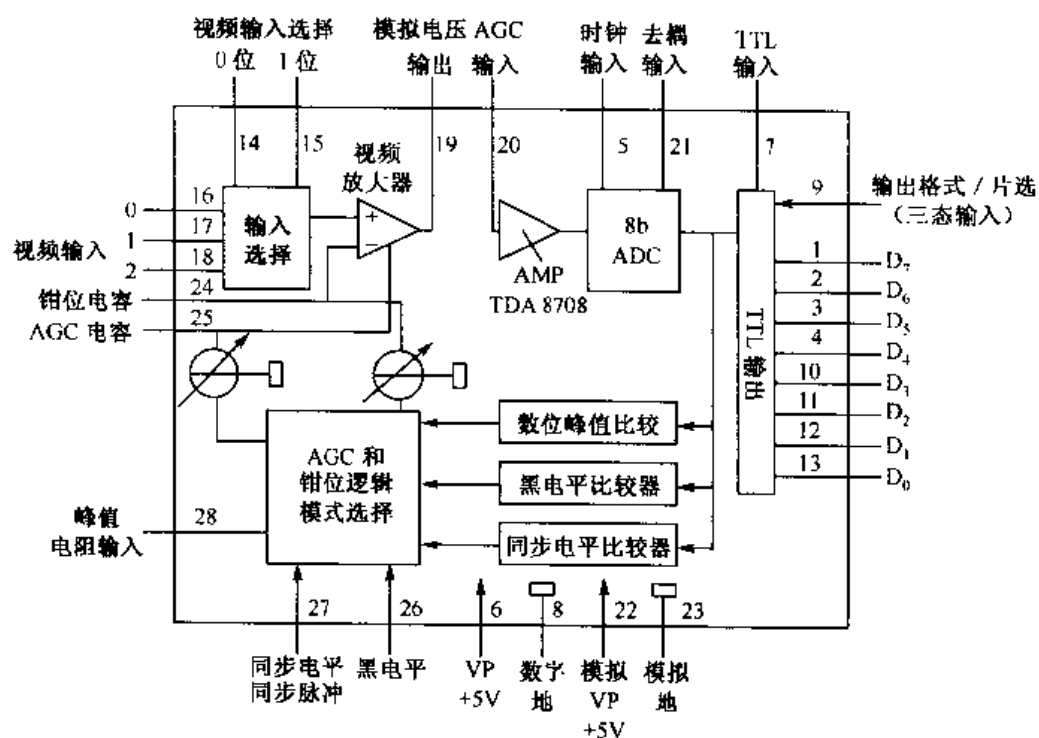


图 3-11 视频模拟输入接口 TDA 8708 原理方框图

表 3-1 YUV411 和 YUV422 资料格式

Y:U:V= 4:1:1					Y:U:V= 4:2:2				
OUT	1	2	3	4	OUT	1	2	3	4
bit0	Y0	Y0	Y0	Y0	bit0	Y0	Y0	Y0	Y0
bit1	Y1	Y1	Y1	Y1	bit1	Y1	Y1	Y1	Y1
bit2	Y2	Y2	Y2	Y2	bit2	Y2	Y2	Y2	Y2
bit3	Y3	Y3	Y3	Y3	bit3	Y3	Y3	Y3	Y3
bit4	Y4	Y4	Y4	Y4	bit4	Y4	Y4	Y4	Y4
bit5	Y5	Y5	Y5	Y5	bit5	Y5	Y5	Y5	Y5
bit6	Y6	Y6	Y6	Y6	bit6	Y6	Y6	Y6	Y6
bit7	Y7	Y7	Y7	Y7	bit7	Y7	Y7	Y7	Y7
bit8	X	X	X	X	bit8	V0	V0	V0	V0
bit9	X	X	X	X	bit9	V1	V1	V1	V1
bitA	X	X	X	X	bitA	V2	V2	V2	V2
bitB	X	X	X	X	bitB	V3	V3	V3	V3
bitC	V6	V4	V2	V0	bitC	V4	V4	V4	V4
bitD	V7	V5	V3	V1	bitD	V5	V5	V5	V5
bitE	V6	V4	V2	V0	bitE	V6	V6	V6	V6
bitF	V7	V5	V3	V1	bitF	V7	V7	V7	V7

SAA7191 不仅产生数字式 YUV 信号,还产生行场同步信号及一些控制信号,如控制 A/D 转换控制器进行消隐电平钳位的 HC 信号,指示同步位置以便进行自动增益控制的 HSY 信号。同时还有离散时间振荡器和数字锁相回路产生的 LFCO (行锁定频率控制) 信号, LFCO

信号送到时钟发生器 SAA7197。时钟发生器 SAA7197 是基于锁相环的同步时钟发生器，在正常工作时能产生输入时钟 LFCD 的两倍频和四倍频的时钟信号，并能在加电时产生 Power-On Reset 信号，其主要作用是配合 SAA7191 或 SAA7199（数字式视频信号编码器）产生所需的行锁定时钟，该时钟是整个编码、译码系统的基础，并给这两个芯片提供加电复位信号。

## 2. 窗口控制器

窗口控制器 82C9001A PC-Video 是 Chips 公司生产的适用于 ISA 总线 PC 机的视频获取、显示用的专用芯片。其内部功能大致可分为 PC 总线接口、视频输入剪裁、变化比例变换，输入窗口 VGA 同步、色键控制以及视频帧内存 VRAM 读、写、刷新控制器。PC-Video 在视频信号获取器中的作用是集总线接口、窗口控制逻辑和内存接口于一体。窗口控制器通过对控制状态寄存器编程可以提供下述功能：

- 在计算机图形监视器上，能够显示全屏幕的活动图像。
- 为显示运动图像，PC Video 能够改变扫描速度，实现窗口控制。
- 通过独立的 X、Y 坐标和彩色键联（Color Key）信号可实现窗口位置控制。
- 真彩色图像的获取和控制。
- 用广播质量的视频带宽，输入分辨率可达 1024×512 像素。
- 支持工业标准视频输入格式，如 NTSC, PAL, SECAM, S-VHS, RGB。
- 支持标准 4:1:1 和 4:2:2 的 YUV 及 16 位 RGB 格式。
- 输出放大因子可为 2、4 和 8。

窗口控制器总体逻辑框图如图 3-12 所示，可分成四部分。

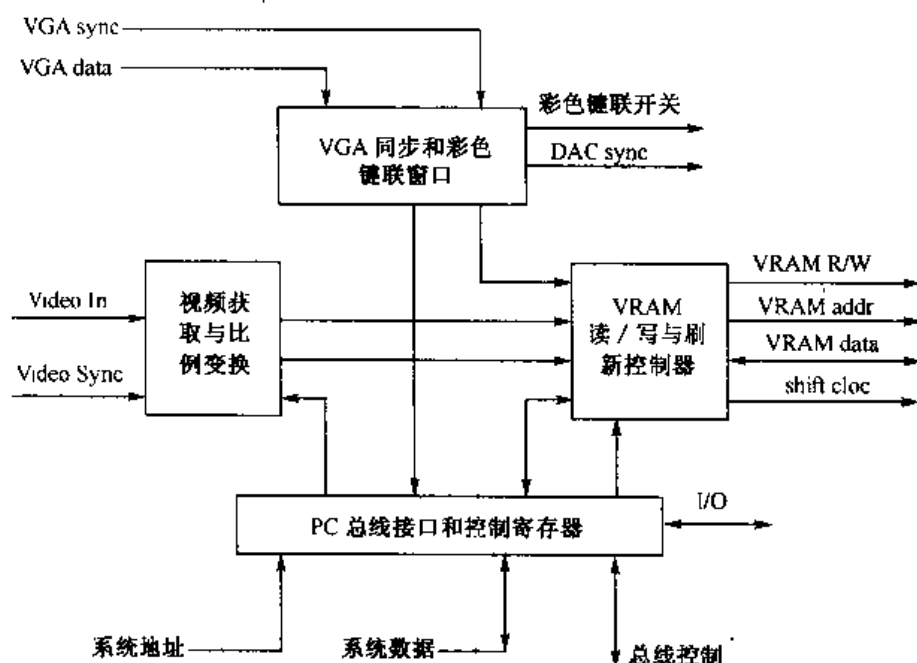


图 3-12 PC Video 窗口控制器

(1) PC 总线接口部分。这部分主要包括 I/O 寄存器地址映像、帧存储地址映像以及帧内存读写功能。主机访问 PC Video 内部功能模块要由这部分控制，同时这部分还包括 I<sup>2</sup>C 总线接口和 4 个用户扩展寄存器选通接口。通过设置寄存器可把 PC Video 的 I/O 地址设在

128 个位置之一帧，内存的线性地址则可位于系统 1MB~15MB 的一个空间上。访问方式为 16 位。

(2) 视频输入剪裁、变比例部分。这部分主要功能是把视频资料按照使用者的定义处理后，送到 VRAM 读写模块中。首先视频资料根据用户的定义确定是选择奇数场或是偶数场，还是选择全帧，紧接着要定义输入窗口的大小和位置。然后再确定是选择捕获窗口外的图像，还是捕获窗口内的图像，或者是选择整个视频图像的有效区域。最后，视频资料可对横向、纵向分别改变比例，改变比例范围为 1/64~64/64。改变比例过程是靠周期性的减少像素和扫描行来实现的。因此，对 YUV422 或 YUV411，RGB444 等做了不同的处理以保证图像内容的正确性。

(3) VRAM 读写、刷新控制部分。这部分主要完成对外接 VRAM 的访问和选体等的控制。按照 PC 总线接口部分、视频输入剪裁、变比例部分，以及输出窗口 VGA 同步、色键控制部分的要求把资料写入或读出帧存储。由于视频获取和 PC 主机读写都是通过 VRAM 的随机读写埠，因此两者不能同时进行。而视频资料输出到 D/A 转换器是通过 VRAM 的串行资料端口，所以，不论是视频获取，还是主机读写，显示输出均可正常工作。此外，在视频资料写入时，还可设置写屏蔽字，对帧内存中的某些位禁止写入。帧存的字宽为 16b，空间分辨率为 1024×512 像素。

(4) 输入窗口 VGA 同步、色键控制部分。这部分主要用来驱动帧内存的资料同 VGA 的视频信号同步读出，并送到 D/A 转换器变成模拟的 RGB 信号，然后通过一个模拟开关与 VGA 视频信号叠加输出到 VGA 监视器上。模拟开关的控制信号可以是：

- 定义一个显示窗口。
- 定义某个彩色键值，当色键值与 VGA 输入资料值相等时，即称色键匹配。

根据上面两项定义的 VGA 屏幕可产生 4 个不同的区域，如图 3-13 所示，可对任意区域选择视频输出或 VGA 输出。图 3-13 中  $F_0$  为非窗口，非色键区； $F_1$  为窗口区，非色键区； $F_2$  为非窗口区，色键区； $F_3$  为窗口和色键共同作用的区域。

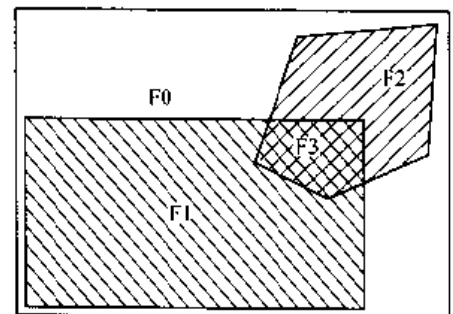


图 3-13 VGA 屏幕叠加效果

### 3. 帧内存系统

PC Video 控制图像的采集和存储，采集资料率高达 14.75MHz，即要求最小访问时间小于 67ns。而设计中选用的 VRAM 是 TC524256Z-10，其访问时间约为 100ns。为解决存储速度问题，采用了双体结构的内存，即顺序的两个像素一次打入两个存储体，使访问周期下降到 134ns。TC524256Z-10 有一个随机埠，还有一个串行访问端口，当输出像素数据时，其串行输出的时钟为像素时钟的 1/2 频率，用锁存器和数据选择器将两个存储体的输出交替送入 D/A 转换器，其原理框图及时序图如图 3-14 所示。图中 Data A 为内存 0 体的输出，DATA B 为内存 1 体的输出，SRCLK 为串行访问口时钟。串行资料在行消隐期间打入内部移位寄存器中。

帧内存的主要作用有三个：

(1) 从摄像机来的视频信号，经过 A/D 变换、数字译码，在视频窗口控制器的控制下，将它们实时地存到帧内存中，大约 74ns 存一个像素资料。

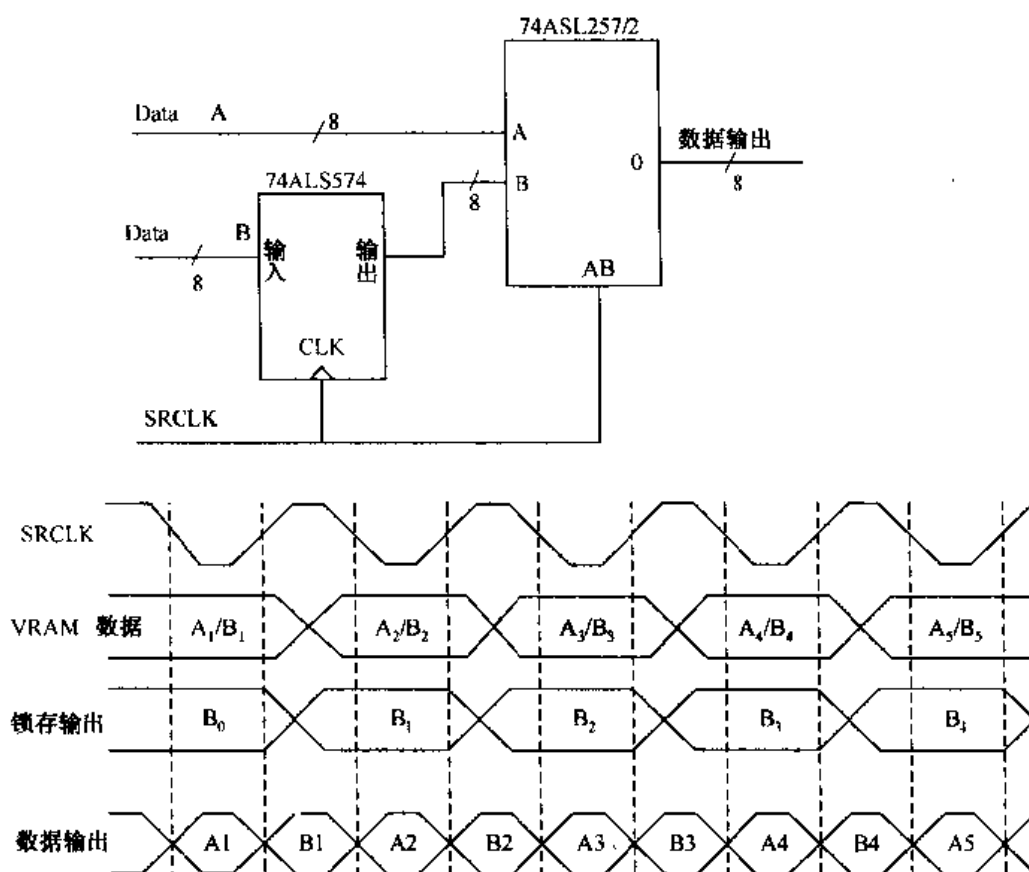


图 3-14 双体内存工作原理

(2) 彩色监视器每隔 74ns 要从帧内存中取一个像素资料(在视频信号正程时), 经 D/A 转换变成模拟的 RGB 信号, 供彩色监视器显示帧内存中真彩色全屏幕运动图像使用。

(3) 计算机可以通过视频窗口控制器对帧内存的内容进行读写操作。帧内存的视频像素信息读到计算机后, 通过编程可以实现各种算法, 完成视频图像处理的任务。同时也可完成帧内存的存盘和读盘的任务。

#### 4. 数模转换和矩阵转换

这部分由两个器件组成, 即 D/A 转换器 SAA9065 和视频信号处理器 TDA4680。

SAA9065 是 YUV 方式的视频 D/A 转换器, 其输入资料为 YUV411 或 YUV422 的视频资料。最高数据率为 30MHz, 内部设有色差信号插值电路和增强图像用的滤波器。输出是 Y、 $-(R-Y)$ 、 $-(B-Y)$  或 Y、 $R-Y$ 、 $B-Y$ , 1V 峰值 (75Ω) 信号。内部功能控制通过 I<sup>2</sup>C 总线实现。

TDA4680 是用于处理亮度和色度信号的模拟电路, 其输入是 Y、 $-(R-Y)$ 、 $-(B-Y)$ , 输出是 R、G、B。TDA4680 本来是作为彩色电视机显像管电子枪的前级, 此处用来实现 YUV 到 RGB 的转换, 以及对亮度、色饱和度、对比度等参数的调整。

YUV 到 RGB 的转换是通过一个 YUV-RGB 矩阵变换网络变成 RGB, 其中矩阵因子因制式不同稍有差别, 目的是补偿信号传输中的失真。

#### 5. 视频信号和 VGA 信号的叠加

视频输出的 RGB 信号和从 VGA 显示卡上引过来的信号是完全同步的, 因为 PC Video 的输出是靠 VGA 的同步信号驱动的, 所以用适当的方法交替地切换两路信号即可实现两路

输出的叠加。由于两路信号均为模拟信号，因此选用了模拟开关电路实现两信号的叠加，具体器件为 74HCT4053，模拟开关的控制信号由 PC Video 给出，控制模拟开关在两路信号间切换，从而实现了输入视频信号在 VGA 监视器上的窗口显示和色键。

### 6. 数字式多制式视频信号编码部分

这部分只选用了数字式多制式视频信号编码器 SA7199。它是以数字方式进行视频信号编码的编码器，支持 PAL 和 NTSC 两种制式，输出有 CVBS 和 Super VHS 两组信号，工作模式有 4 种，可借助软件利用 I<sup>2</sup>C 总线进行控制。4 种模式的主要区别是同步信号的产生方式，即同步信号可由编码器根据自身的时钟产生，也可由外部电路产生，或者还可以由编码器和外部信号同步产生。输入资料可是 YUV，也可能是 RGB。SAA7199 内部设有 3×256 字节的查找表，用于  $\gamma$  校正等查找表运算，比如对比度、色调等运算，该查找表也可跳过不用。若输入资料为 RGB，要经过一个内部转换矩阵变成 YUV，送到内部全数字方式工作的编码器中进行编码，编码后的数字信号经内部 D/A 转换后输出模拟视频信号。

### 3.2.2 彩色全电视信号的数字锁相和数字译码

在彩色全电视信号模拟锁相和译码电路中，一般采用模拟式锁相回路跟踪输入信号的行频，其工作原理如图 3-15 所示：PD 是相位检测器，它检测压控振荡器（VCO）输出的系统时钟和输入行同步之间的相位差，经过低通滤波器（LF），或称校正环节，能改善锁相系统的品质指针，提高锁相精度，改善相位锁定时间。低通滤波器的输出给压控振荡器，改变振荡频率，经过分频器（/N）使其与给定行频一致，完成锁相任务。

在数字式锁相和数字式译码电路中用离散时间振荡器 DTO（Discrete Time Oscillator）代替压控振荡器（VCO）。DTO 的工作原理如图 3-16 所示。DTO 的核心部件是加法器，加法器的输出通过一个寄存器反馈到输入端，和另一个增量 P 在每个时钟的上升沿相加，使得  $X_n$  为线性增长序列。当  $X_n$  超过加法器的最大值  $q$  时，产生溢出，忽略进位，则又产生由小到大的序列，如图 3-17 所示。

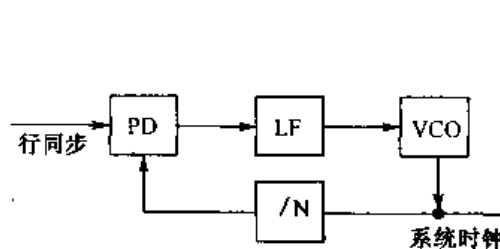


图 3-15 模拟锁相电路图

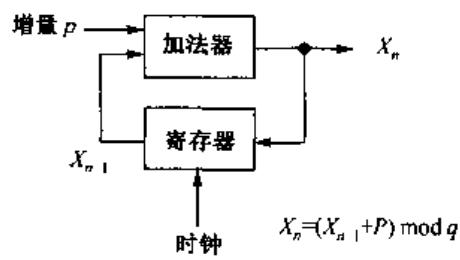


图 3-16 离散时间振荡器

加法器的输出为：

$$X_n = (X_{n-1} + P) \bmod q \tag{3-17}$$

由上式可知， $X_n$  是一个离散时间锯齿信号，时间间隔  $T_0 = 1/f_0$ 。由图 3-17 可知， $p$  和  $q$  的比例与时间  $T$ 、 $T_0$  的比例相等，即

$$\frac{p}{q} = \frac{T}{T_0} = \frac{f_0}{f_T} \tag{3-18}$$

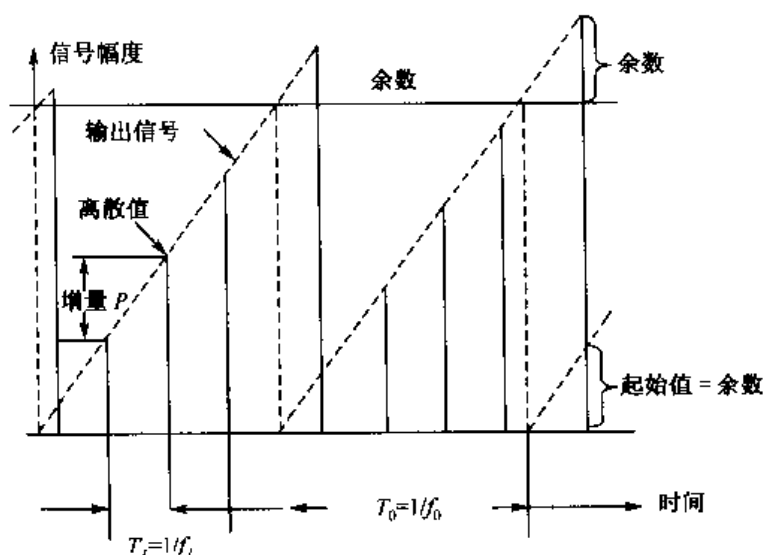


图 3-17 离散时间震荡的输出信号

若将  $q$  归一化为 1, 则有

$$f_0 = pf_T \quad (3-19)$$

因此, 离散时间振荡器的输出频率  $f_0$  受  $p$  控制, 像模拟锁相回路一样, 输入频率为  $f_T$  的信号会产生输出频率  $f_0 = pf_T$  的信号。

彩色全电视信号数字锁相和译码的具体电路和工作原理如图 3-18 所示。数字式彩色全电视信号首先经过一个带通滤波器, 把色差信号滤出, 最终得到 UV 信号, 另外一路经过彩色副载波陷波电路, 去掉色差信号, 得到 Y 信号。再经过数字式同步分离电路得到输入信号的行同步, 通过检相器检测输入行频和系统时钟, 经分频后得到的行频两者之间的相位差, 再经过平滑滤波得到增量  $p$ , 由式 (3-19) 可知, 它由  $p$  线性控制离散时间振荡器得到系统时钟信号  $f_0$ , 这就是数字锁相的工作原理, 具体电路框图如图 3-18 左半部所示。

数字译码电路, 在模拟电路中, 彩色信号采用正交调幅方法进行调制, 彩色全电视信号的表达式为:

$$CVBS = Y + U \sin \omega_{sc} t + V \cos \omega_{sc} t \quad (3-20)$$

其中  $Y$  为亮度信号,  $UV$  为色差信号,  $\omega_{sc}$  为彩色副载波频率, 用带通滤波器可分离出  $C$  信号:

$$C = (R - Y) \sin \omega_{sc} t + (B - Y) \cos \omega_{sc} t \quad (3-21)$$

再用一个具有同样频率和相位的副载波信号乘以  $C$  得:

$$\begin{aligned} C \sin \omega_{sc} t &= (R - Y) \sin \omega_{sc} t \cdot \sin \omega_{sc} t + (B - Y) \cos \omega_{sc} t \cdot \sin \omega_{sc} t \\ &\approx (R - Y) - (R - Y) \cos 2\omega_{sc} t \end{aligned} \quad (3-22)$$

用低通滤波器滤掉  $(R - Y) \cos 2\omega_{sc} t$ , 即可恢复  $R - Y$  分量, 同样用  $\cos \omega_{sc} t$  乘  $C$ , 即可恢复  $B - Y$  分量。具体电路如图 3-18 右半部所示, 也采用数字锁相的办法产生副载波频率, 将它作为地址去找 ROM 中  $\sin, \cos$  表, 读出的结果和输入的数字彩色信号相乘, 就达到了译码的目的。

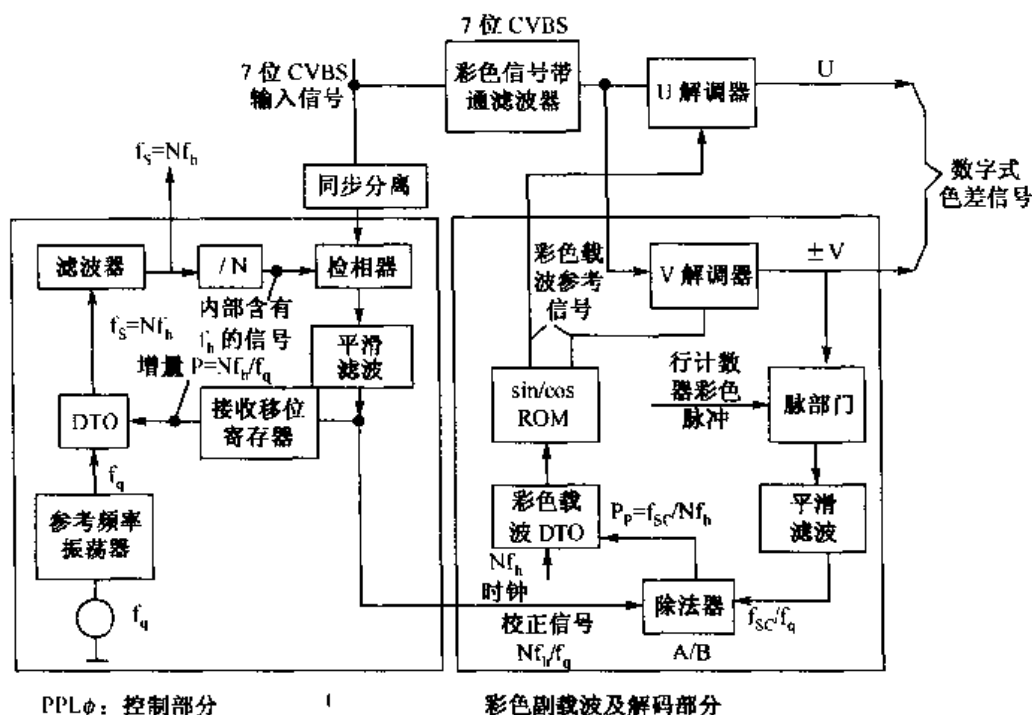


图 3-18 数字锁相与数字译码原理框图

在图 3-18 的右半部，数字式彩色全电视信号首先经过副载波滤波器，得到  $U$ 、 $V$  的调制信号，再经过解调，即分别乘以  $\sin \omega_{sc} t$  和  $\cos \omega_{sc} t$  得到  $U$ 、 $V$  分量。在此电路中，数字锁相电路是将  $V$  的解调信号经过脉冲门和平滑滤波得到  $A = f_{sc} / f_q$ （其中  $f_{sc}$  是彩色副载波频率， $f_q$  是参考时钟频率），如前所述，另一路  $B = Nf_H / f_q$ （其中  $f_H$  是行频， $N$  是分频器值，对于 PAL 制， $N = 864$ ，对于 NTSC 制， $N = 858$ ），经过 A/B 除法器，得到离散时间振荡器的增量  $p = f_{sc} / Nf_H$ ，控制 DTO 的输出。DTO 的输出作为地址，到 ROM 中查  $\sin$  和  $\cos$  值，与从数字式彩色全电视信号分离的  $U$ 、 $V$  信号相乘，再经过低通滤波器就得到了数字式  $U$ 、 $V$  信号。数字式彩色全电视信号如式 (3-16) 所示，经过陷波器滤掉彩色副载波分量，去掉噪声，经过校正就能得到数字式  $Y$  信号。

### 3.3 视频信号的采集

只有将视频信号采集到计算机中，才可以对它进行编辑和处理。视频信号的采集又可以分为模拟视频和数字视频的采集。采集模拟视频需要硬件设备的支持，采集数字视频实际上是一种格式的转换。

#### 3.3.1 采集模拟视频

通常我们借助视频采集卡来完成模拟视频的采集工作。首先在采集界面上，通过设置选项，来决定是采用 AV 端子还是 S 端子来输入视频信号。然后，将图 3-19 中的摄像机的 AV 端子和视频采集卡的 AV 端子连接起来，或者将摄像机的 S 端子和视频采集卡的 S 端子连接起来。完成了连接的准备工作后，就可以一边播放视频一边采集视频了。



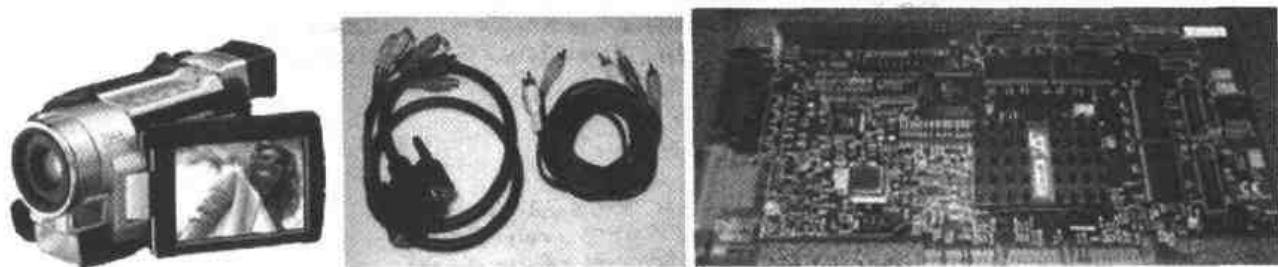


图 3-19 用 AV 端子或 S 端子连接视频卡和摄像机

### 3.3.2 采集数字视频（VCD 转换和 HyperCam）

通常情况下，当我们没有视频卡时，可以将 VCD 视频转换为视频编辑可以使用的格式，也可以利用 HyperCam 软件实时捕捉计算机屏幕上播放的视频图像。

#### 1. 用超级解霸中的“VCD 转 AVI”功能采集

利用超级解霸中的“VCD 转 AVI”功能，可以将 DAT 格式的视频文件转换为 Premiere 可以编辑的视频文件格式。“VCD 转 AVI”的界面如图 3-20 所示。

首先单击【视频文件】按钮选择 VCD 视频文件，然后播放该视频找到需要采集的开始，停止播放视频。单击【开始压缩】按钮就可以进行采集或转换了，单击【停止压缩】按钮，就完成了数字视频的采集。

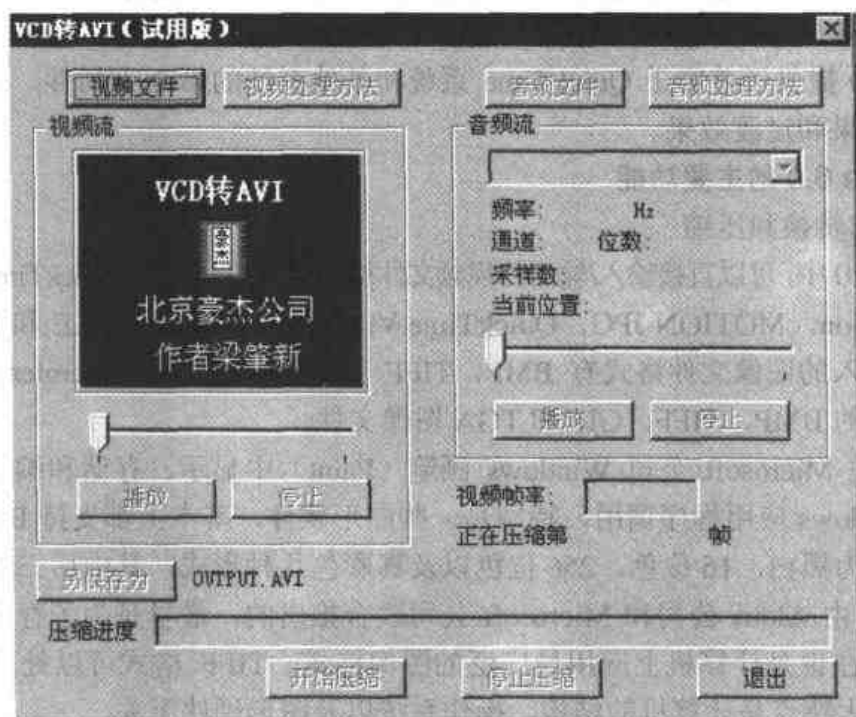


图 3-20 “VCD 转 AVI” 界面

#### 2. 用 HyperCam 软件

使用 HyperCam 软件可以实时捕捉计算机屏幕上播放的视频图像，利用这种方法可以采集游戏的视频画面。HyperCam 软件的界面如图 3-21 所示。

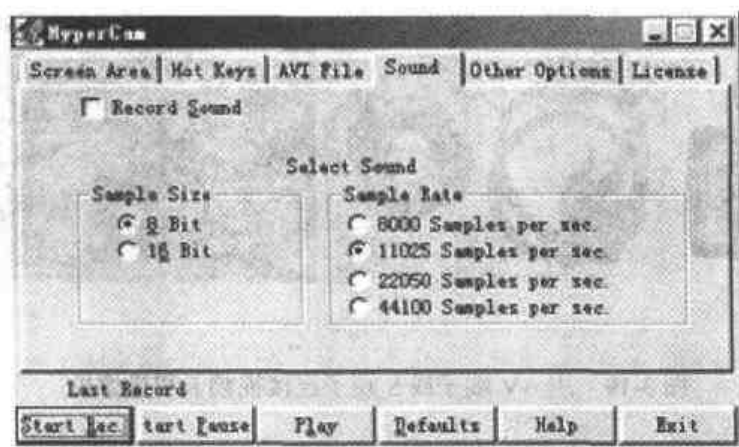


图 3-21 HYPERCAM 软件的界面

## 3.4 Premiere 简介

### 3.4.1 功能与特色

Premiere 6.0 是 Adobe 公司推出的 Premiere 系列软件的第 6 个版本。从 5.0 以后的版本都同时支持 Windows 95/98、Windows NT、Windows 2000。Premiere 6.0 为视频节目的创建和编辑提供了更强大的支持,在进行视频编辑、节目预览、视频捕获以及节目输出等操作时,可以在兼顾视频效果和播放速度的同时,实现更佳的影音效果。

Premiere 6.0 提供了兼容于 QuickTime 系统和其他系统的第 3 方插件,使用这些插件,可以实现视频效果和过渡效果。

#### 1. Premiere 6.0 的主要功能

##### (1) 素材的转换和压缩

在 Premiere 6.0 中,可以直接输入/输出的视频文件格式有 AVI, QTM (QuickTime for Windows), Autodesk Animation, MOTION-JPG, QuickTime Video, Raw Indeo, RLE 和 Indeo3.2 等。

可以直接输入的图像文件格式有 BMP, TIFF 和 TGA 格式等。Premiere 还可以将视频项目输出成连续的 BMP, TIFF, GIF 和 TGA 图像文件。

BMP 格式是 Microsoft 公司 Windows 画笔 (Paint) 中显示、存储和编辑的图像格式,可以被多种 Windows 应用程序调用,特别是一些图形软件,基本上都支持 BMP 格式。BMP 格式可以简单分为黑白, 16 位色, 256 位色以及真彩色几种形式,其中后三者有彩色映像。

TIFF 格式是由 Aldus 公司和 Microsoft 公司联合推出的,最早是为了存储扫描图像而设计的,因而它现在也是计算机上应用最广泛的图像格式。TIFF 格式可以处理黑白、灰度、彩色图像,其最大特点是计算机的结构、操作系统以及图形硬件无关。

TGA 格式是由 Truevision 公司出品的,已经被国际图形制作业所接受,现在已经成为数字化图像,运用光线跟踪算法所产生的常用图像格式。

GIF 格式即图形交换格式,它形成一种压缩的 8 位图像文件。这种格式的文件,目前多用于网络传输。

##### (2) 视频/音频捕捉和剪辑

Premiere 6.0 支持视频和音频的捕捉，同时还可以压缩和剪辑捕捉后的素材。

### (3) 视频编辑功能

Premiere 6.0 在 Timeline 窗口中处理视频片段，最多支持 99 个 Video 轨道。

### (4) 丰富的过渡效果

过渡效果又叫转场效果，可以方便地在各个视频片段之间实现丰富多彩的过渡。

### (5) 添加运动效果

通过视频片段添加运动效果，使制作出的视频作品极富动感。

### (6) 对 Internet 的支持

Premiere 6.0 添加了 RM (Real Media) 这一网络主流视频格式的输出功能，使得我们可以直接在 Premiere 6.0 中将作品输出成 RM 格式。换句话说，对 RM 格式视频输出的支持是 Premiere 系列产品是一个飞跃，紧紧跟上 Internet 网络发展的步伐。

## 2. Premiere 6.0 的新增功能

### (1) 新增的网络输出属性

增加了 Advanced ReelMedia Export 功能模块，添加了 RM (RealMedia) 这一网络主流视频格式的输出功能。

增加了 Advanced Windows Media Stream Web Export (高级 Windows 媒体流的 web 输出) 功能模块。

### (2) 新增的 Audio Mixer (混音器)

通过新增的 Audio Mixer，可以得到音频的混频效果。

### (3) History (历史) 面板

通过这个工具，在视频处理的时候，可以自由撤销或重做几乎所有的操作。

### (4) 新增的 Effect Controls (效果控制) 面板

使用此面板，可以控制在视频片段中添加的所有效果，包括运动效果以及视频效果。

### (5) 新增的 Workspace (工作区) 管理功能

在 Window 菜单中添加了 Workspace 级联菜单，在这里可以按照视频制作需求和使用熟练程度选择几种常用的工作区模式。也可以将当前的工作区的模式保存起来，以便日后调用。

### (6) 更新后的视频/音频滤镜

Premiere 6.0 中，视频 (滤镜) 效果增加到了 70 多种，并且放弃了滤镜 (Filter) 的称谓，改为 Video Effect (视频效果) 和 Audio Effect (音频效果)。同时增添了 Video 面板和 Audio 面板。

### (7) Project 窗口中全新的 Bin (箱) 概念

全新概念的 Bin (箱) 使得素材的组织和管理更容易。

### (8) 菜单功能的添加和菜单项的更新

1) 增加了 Timeline 菜单，将其他菜单中与 Timeline 窗口操作有关的菜单项统一安排到这里。

2) 加了片段的“智能再制” (Duplicate Clip) 功能，并且将这种好用的新功能放置在 Edit 菜单中。

3) 将 Preferences (参数选择) 级联菜单从 File 菜单搬到了 Edit 菜单中。

4) 将 Marker(标记)详细分成了 Clip Marker(片段标记, 在 Clip 菜单中)和 Timeline Marker (Timeline 窗口标记, 在 Timeline 窗口中), 分别设置于不同级别的对象上。

5) Window 菜单中添加了 Workspace 级联菜单和 Window Options 级联菜单, Window Options 级联菜单中包含 4 个菜单项, 分别用来设置 Project 窗口、Timeline 窗口、Monitor 窗口以及 Audio Mixer 窗口的属性。

6) Clip 菜单中添加了 Advanced Options 级联菜单, 用来设置视频的一些高级选项。

### 3. Premiere 界面的组成

#### (1) Premiere 界面


进入 Premiere 6.0 后的界面如图 3-22 所示。屏幕上方是菜单栏, 含 7 个菜单项; 另外还有若干个窗口, 有 Project (项目) 窗口、Monitor (监控) 窗口、Timeline (时间线) 窗口和 Transitions (过渡)/Video/Audio 窗口等。





图 3-22 Premiere 的界面

#### (2) Project 窗口

Project (项目) 窗口用来管理当前项目中用到的各种素材。所有要编辑和处理的素材都要放到此窗口中。素材有 3 种显示模式: Icon View 模式、Thumbnail View 模式和 List View 模式。

1) Icon View 模式。当窗口下部的  图标被按下时, Project 窗口处于图标显示模式, 如图 3-23 所示。

2) Thumbnail View 模式。单击  按钮后, 可以切换到缩略图显示模式。如图 3-24 所示。

3) List View 模式。单击  按钮, 可以切换到列表显示模式。如图 3-25 所示。


当然, 也可以通过在 Project Window Options 对话框选择不同的选项来自定义显示模式。单击 Project 窗口标题栏最左边的  控制按钮, 从弹出的菜单中选择 Options, 可以打开 Project Window Options 对话框, 如图 3-26 所示。



图 3-23 Icon View 模式

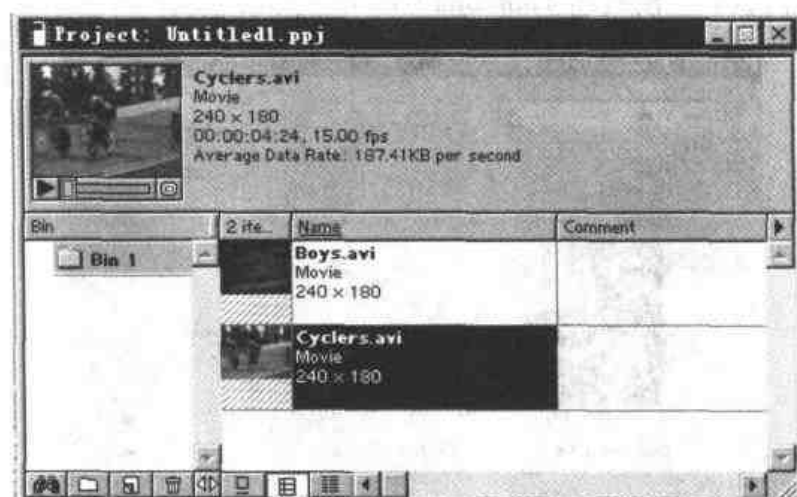


图 3-24 Thumbnail View 模式

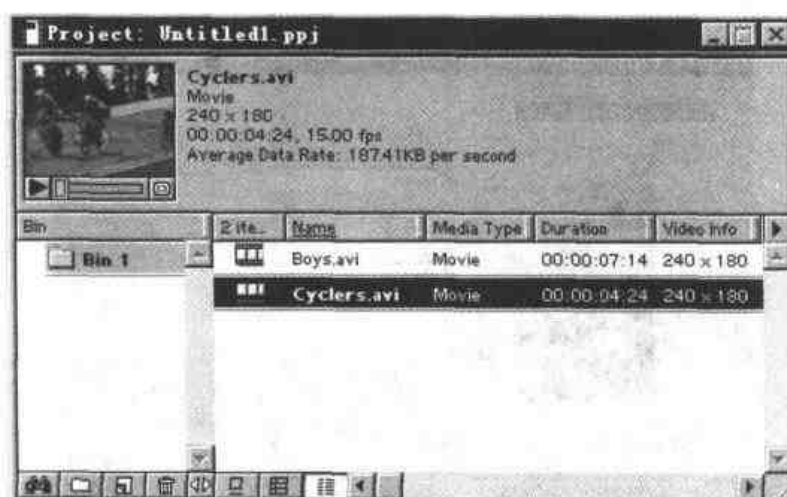


图 3-25 List View 模式

List View（列表模式）可以在 Fields（字段）设置区指定是否需要显示列出的字段，可选名字和日期等。在 Sortings（排序）设置区域指定排序字段。

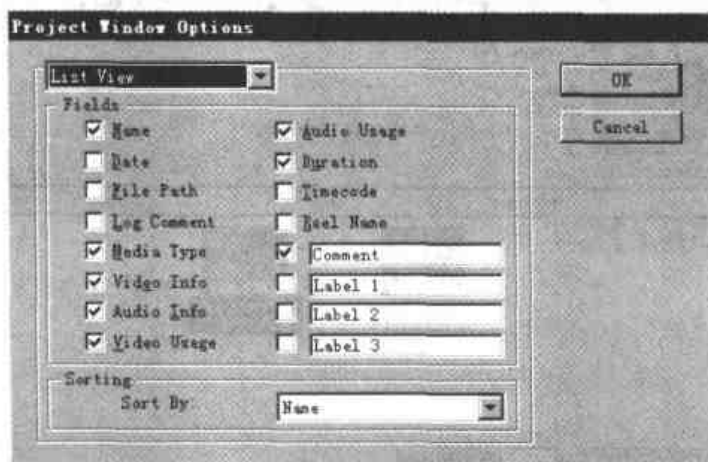


图 3-26 Project Window Options 对话框

Icon View（图表显示）模式和 Thumbnail View（列表显示）如图 3-27 和图 3-28 所示。

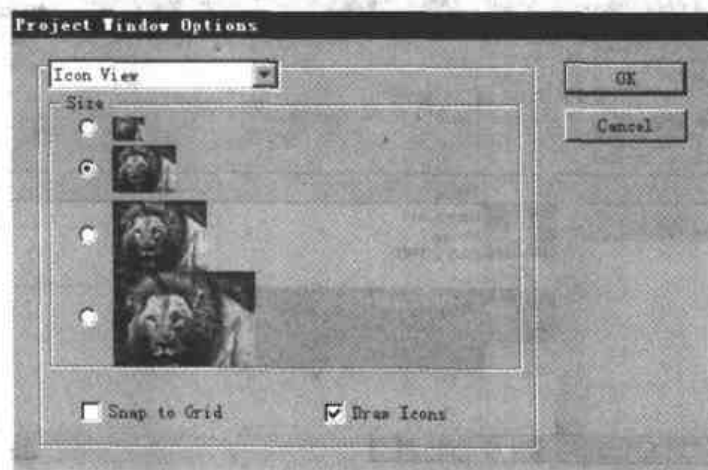


图 3-27 Icon View（图表显示）模式

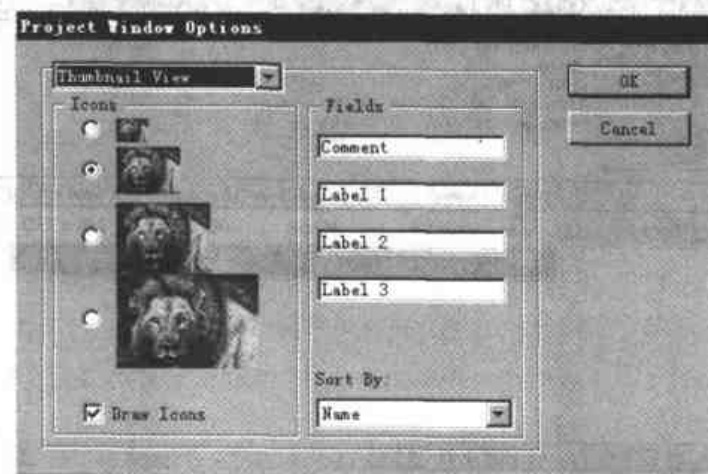


图 3-28 Thumbnail View（列表显示）模式

### （3）Timeline 窗口

Timeline 窗口是一个基于时间轴的显示窗口，如图 3-29 所示。

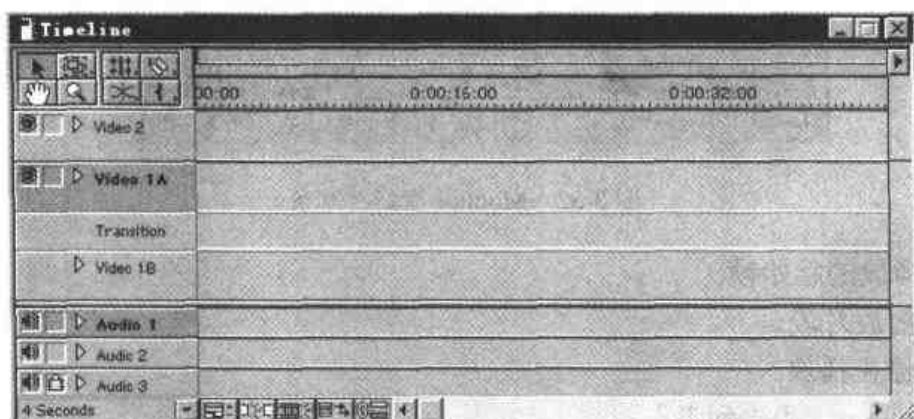


图 3-29 Timeline 窗口

从图 3-29 中可以看到，Timeline 窗口中有视频轨道（Video Track）、音频轨道（Audio Track）以及用来添加过渡效果的过渡轨道（Transition Track）。此外，还有工具条、滑动条和时间坐标轴（最长可设置为 3 小时）等组件。

也可以自己定义 Timeline 窗口的显示模式。单击窗口右上方的按钮，可以打开如图 3-30 所示的窗口控制菜单，选择“Track Options”命令即可自行定义 Timeline 窗口的显示模式。



图 3-30 选择“Track Options”命令

#### （4）Monitor 窗口

Monitor 窗口可以用来播放某个素材片段或者播放整个视频节目，如图 3-31 所示。

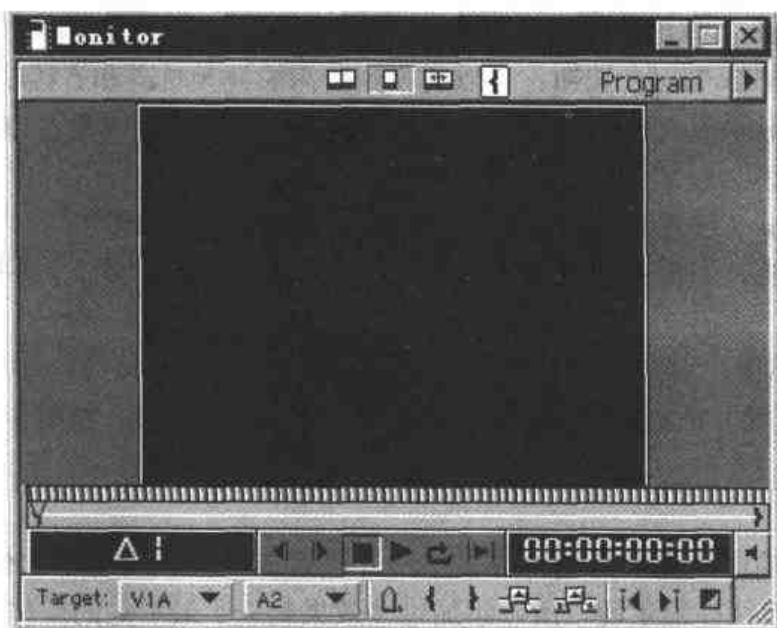


图 3-31 Monitor 窗口

Monitor 窗口控制条如图 3-32 所示，它们的功能分别是：



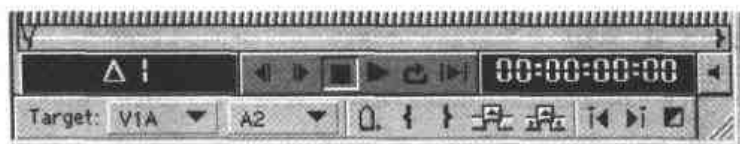


图 3-32 Monitor 窗口控制条

△：定位播放的起始帧。

■ Play：播放片段。

■ Stop：停止播放。

■ Play in to out：播放到出点。

■ Loop：在片段的入点和出点之间循环播放。

■ Frame Back：将画面逐帧往后移动。

■ Frame Forward：将画面逐帧向前移动。

■ Mark In：将片段的当前位置设置成入点，时间监控框的相应位置出现“{”符号。

■ Mark Out：将片段的当前位置设置成出点，时间监控框的相应位置出现“}”符号。

■ Insert：将当前片段插入到编辑线位置，其他在时间上重叠的片段都往后退。

■ Overlay：将当前片段插入到编辑线位置，其他在时间上重叠的片段的重叠部分都被覆盖。

■ Lift：将当前片段从编辑轨道上移走，与之相邻的片段不改变位置。

■ Extract：将当前片段从编辑轨道上移走，接在后面的其他片段的位置被提前。

■ Previous Edit：将编辑线定位于时间轴上前一个片段的开始位置。

■ Next Edit：将编辑线定位于时间轴上后一个片段的开始位置。

■ Set Volume：设置音量。

■ Add Default Transition：在过渡轨道的编辑线位置上加入默认的过渡。

■ Marker Menus：弹出快速操作菜单，可以标记、跳转到或者清除片段。

■ Target 按钮：可以切换到所需的视频轨道和音频轨道。

■ 窗口显示模式选择按钮：从左到右依次可以切换到双重窗口模式、单一窗口显示模式和整理模式。

#### (5) Transitions (过渡) / Info (信息) / Navigator (导航) 面板组

Navigator (导航) 面板：使用不同的颜色条表示 Timeline 窗口中不同的素材片段。如图 3-33 所示。

Info (信息) 面板：用来显示当前选取片段或者过渡效果的相关信息。如图 3-34 所示。

Transitions (过渡) 面板：列出了 74 种可以使用的过渡效果。如图 3-35 所示。

### 3.4.2 过渡效果

一个好的完整的影片通常是由许许多多的片段组成的。在没有添加任何过渡效果之前，这些片段之间的过渡往往很生硬。所以添加过渡效果就成了最常见的操作之一了。

过渡效果共有 74 种，主要有：3D Motion, Dissolve, Zoom, Special Effects, Wipe 等不同类型的过渡效果。

例如，我们为两个片段添加 3D Motion 中的 Motion 过渡效果。



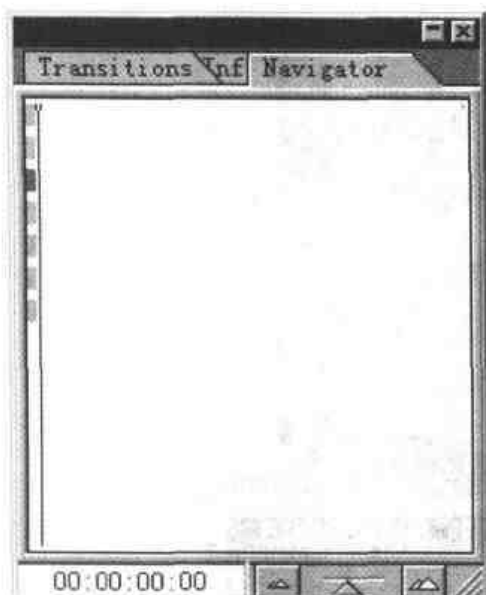


图 3-33 Navigator (导航) 面板

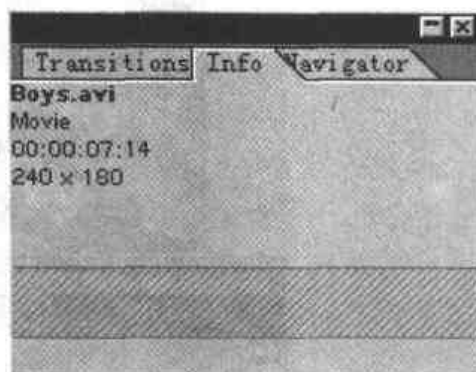


图 3-34 Info (信息) 面板

首先, 将两个片段拖拽到视频 1A 和 1B 轨道中, 让两个片段中间有一个重合区域, 然后选择 3D Motion 中的 Motion 过渡效果, 将它拖拽到过渡轨道中。双击面板中 Motion 过渡效果, 打开 Motion Settings 对话框, 选择“Show Actual Sources”可以显示真实的素材, 还可以选过渡效果开始和结束的位置, 如图 3-36 所示。

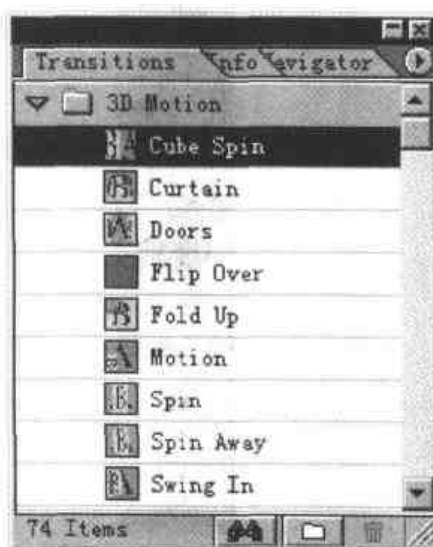


图 3-35 Transitions (过渡) 面板

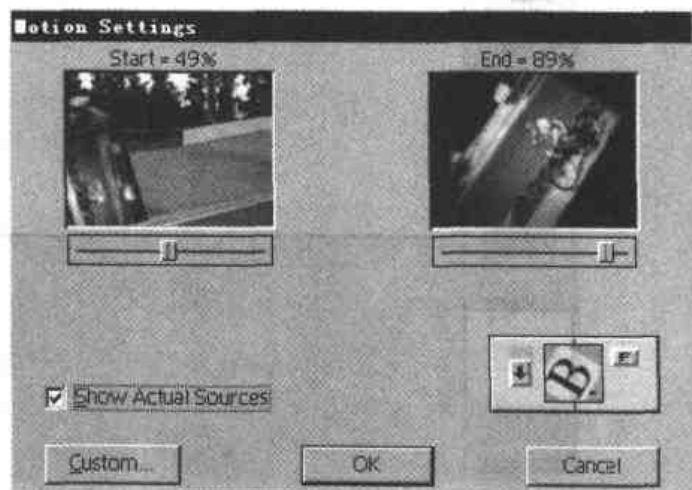


图 3-36 Motion Settings 对话框

Motion 过渡效果是后一个片段从前一个片段后面飞出来, 如图 3-37 所示。

### 3.4.3 字幕和叠加

#### 1. 字幕

Premiere 6.0 提供了功能强大的 Title 窗口, 也叫字幕工作区。选择 File/New /Title 菜单命令, 可以打开字幕编辑窗口, 如图 3-38 所示。

位于左边的是工具栏, 如图 3-39 所示。



图 3-37 Motion 过渡效果

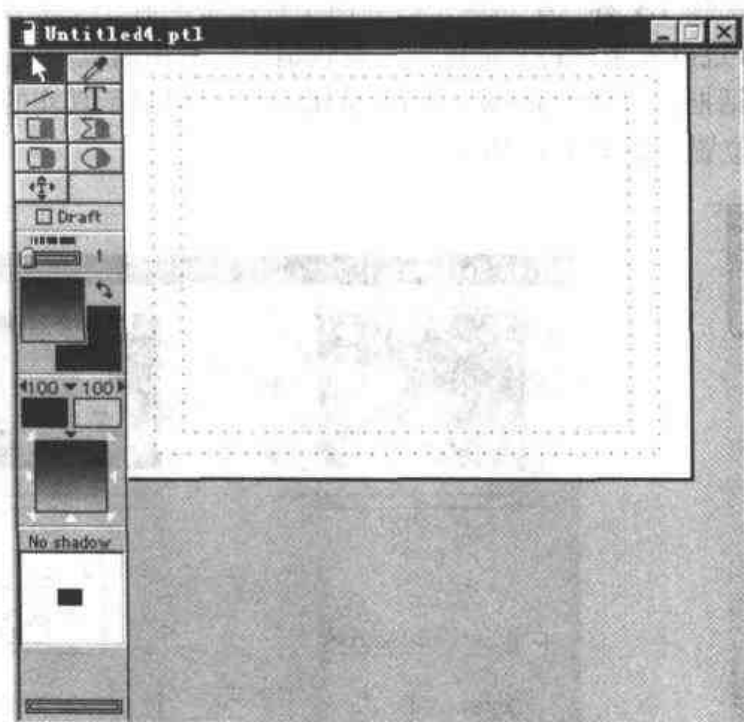








图 3-38 字幕编辑窗口





图 3-39 工具栏


工具栏中的工具有：


-  **Selection Tool (选择工具)**：一般用来选定已经制作完成的字幕文字。
-  **Eyedropper Tool (吸管工具)**：主要用来在标题字幕的文本之间复制颜色。
-  **Line Tool (直线工具)**：用来在字幕窗口中绘制直线。
-  **Type Tool (文字工具)**：用来在字幕标题中添加文字，还可以进行编辑。
-  **Rectangle Tool (矩形工具)**：用来绘制矩形。
-  **Polygon Tool (多边形工具)**：用来绘制多边形。



 **Round Polygon Tool**（圆形矩形工具）：用来绘制圆角矩形。




 **Oval Tool**（椭圆工具）：绘制椭圆。


 **Rolling Title Tool**（滚动字幕工具）：制作滚动效果，同时可以给滚动的文本设定滚动方向、速度和定时等属性。

 **Draft** 显示质量开关。

 **Line Width**（线宽控制工具）：可以改变线条宽度。

 **Object Color/Shadow Color**（对象颜色/阴影颜色设置工具）：单击可以转换设置好的对象和阴影颜色。

 是颜色/透明度设置工具，可以用来设置所选对象或者阴影的渐变色和透明度。单击 **Start Transparency** 可以改变起始颜色的透明度；单击 **End Transparency** 可以设置渐变的总体透明度。

 **Shadow Position**（阴影位置控制工具）：显示对象和阴影之间的相对位置。

## 2. 叠加效果

在 **Premiere 6.0** 中，通过为上面轨道中的片段设置透明度，可以将两个片段叠加起来，同时下面轨道中的片段通过前景片段的透明部分浮现出来。

执行 **Clip/Video/Transparency** 命令，打开 **Transparency Settings** 对话框，如图 3-40 所示。

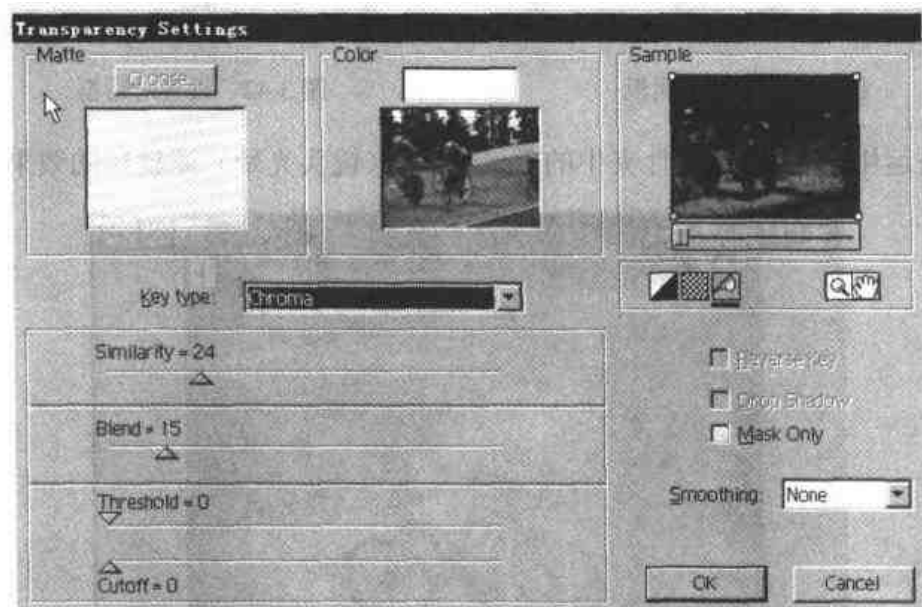


图 3-40 Transparency Settings 对话框

在这个对话框中，通过选择 **Key type** 的类型，可以得到不同的叠加效果。

### 3.4.4 为视频添加滤镜效果

使用 **Premiere 6.0** 中的视频滤镜处理视频，可以得到许多非常精美的特技画面，如风吹、

扭曲等，许多原始素材由于某些原因会有缺陷，通过添加适当的视频滤镜效果，对视频进行处理，可以得到满意的画面。

Premiere 6.0 中的滤镜特技产生的是动态效果，可以随时间的变化而变化。同时，Premiere 6.0 中的一部分滤镜也可以应用于音频。

在 Premiere 6.0 中，一共有多达 73 种视频滤镜。

执行 Window/Show Video Effects 菜单命令，可以看到 Video 面板，如图 3-41 所示。我们可以随意选择某个滤镜，然后将它拖拽到 Timeline 窗口的素材上即可。

此时，会打开一个设置对话框，可以根据需要选择参数，如图 3-42 所示。

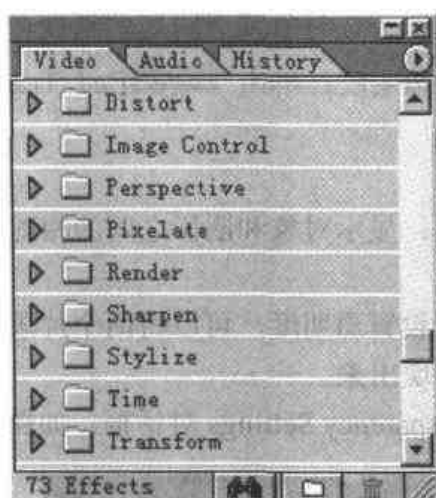


图 3-41 Video 面板

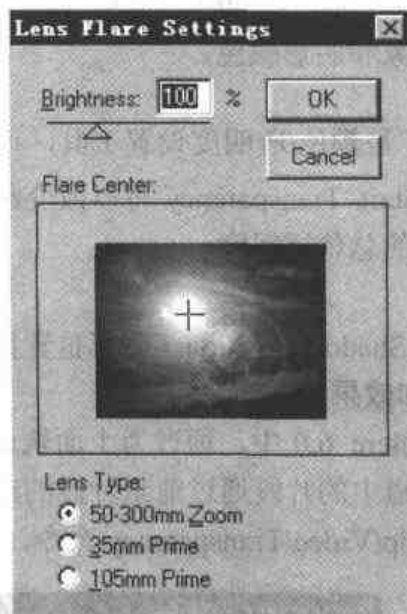


图 3-42 设置对话框

图 3-43 是运用了 **Render** 文件夹中的 **Lens Flare**（镜头光晕）滤镜后的效果。



图 3-43 使用 Lens Flare（镜头光晕）滤镜

### 3.4.5 运动效果

Premiere 6.0 作为视频的编辑和处理软件,可以轻松的制作出动感十足的多媒体作品。运动是多媒体的灵魂,灵活运用动画效果,可以使得视频作品更加丰富多彩。

执行 Window/Show Effect Controls 菜单命令,可以打开 Effect Controls 面板,如图 3-44 所示。单击该面板可以打开 Motion Settings 对话框,如图 3-45 所示。在对话框中可以设置片段的初始位 Start, 路径, 运动的速度和图像的放缩等属性。

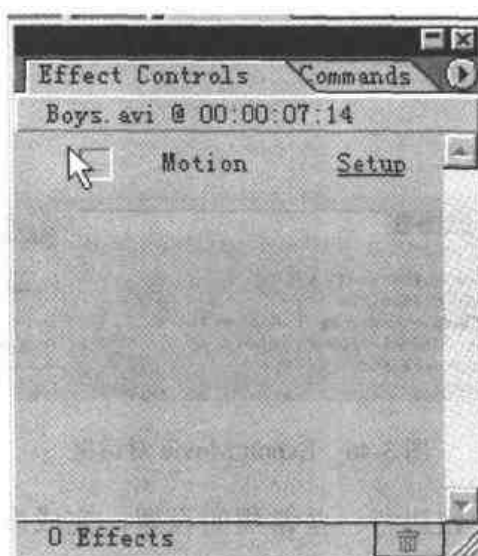


图 3-44 Effect Controls 面板

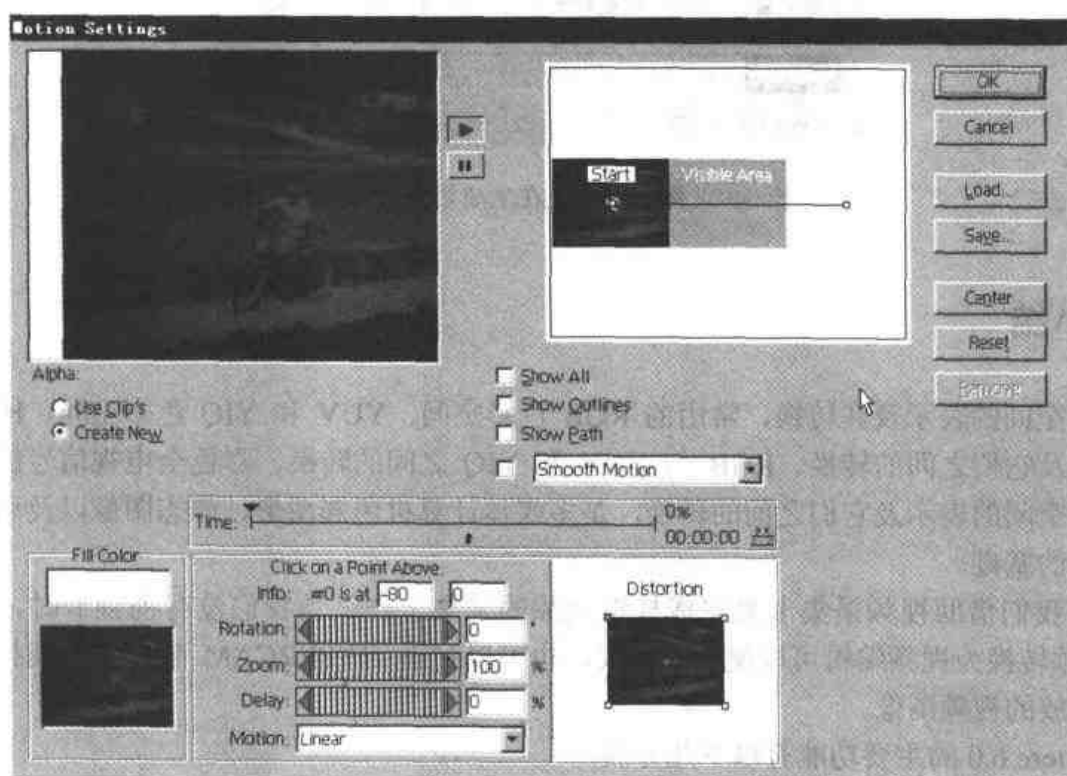


图 3-45 Motion Settings 对话框

### 3.4.6 生成电影

视频项目编辑完成后, 执行 File/Export Timeline/Movie 菜单命令, 进入 Export Movie 对话框, 如图 3-46 所示。命名后即可生成 \*.avi 的电影。

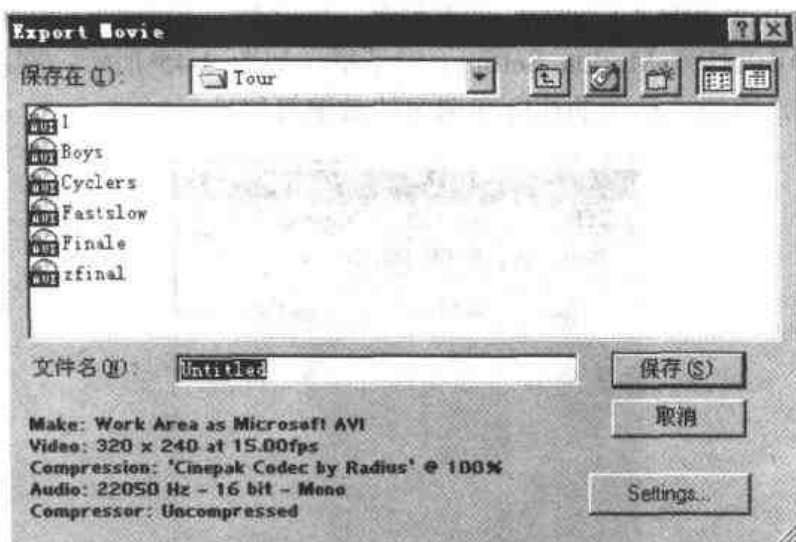


图 3-46 Export Movie 对话框

可以单击【Settings...】按钮进行一些常规的设置, 完成设置后单击【保存】按钮, 就进入了生成阶段, 如图 3-47 所示。



图 3-47 正在生成影片

## 3.5 小结

色彩空间的表示及其转换, 常用的 RGB 色彩空间、YUV 和 YIQ 色彩空间、HIS 色彩空间, 以及它们之间的转换、RGB 与 YUV 和 YIQ 之间的转换、彩色全电视信号的组成, 掌握彩色空间的表示及它们之间的转换, 是多媒体计算机色彩图形、静态图像以及动态图像处理算法的基础。

通常我们借助视频采集卡来完成模拟视频的采集工作。当我们没有视频卡时, 可以将 VCD 视频转换为视频编辑可以使用的格式, 也可以利用 HYPERCAM 软件实时捕捉计算机屏幕上播放的视频图像。

Premiere 6.0 的主要功能有以下几方面。

过渡: 一个好的完整的影片通常是由许许多多的片段组成的。在没有添加任何过渡效果之前, 这些片段之间的过渡往往很生硬, 所以添加过渡效果就成了最常见的操作之一了。

叠加：在 Premiere 6.0 中，通过为上面轨道中的片段设置透明度，可以将两个片段叠加起来，同时下面轨道中的片段通过前景片段的透明部分浮现出来。

滤镜：使用 Premiere 6.0 中的视频滤镜处理视频，可以得到许多非常精美的特技画面，如风吹、扭曲等，许多原始素材由于某些原因会有缺陷，通过添加适当的视频滤镜效果，对视频进行处理，可以得到满意的画面。

运动：Premiere 6.0 作为视频的编辑和处理软件，可以轻松的制作出动感十足的多媒体作品。运动是多媒体的灵魂，灵活运用动画效果，可以使得视频作品更加丰富多彩。

### 3.6 习题

1. 国际上常用的视频制式有\_\_\_\_\_。  
(1) PAL 制 (2) NTSC 制  
(3) SECAM 制 (4) MPEG  
(A) (1) (B) (1) (2)  
(C) (1) (2) (3) (D) 全部
2. 全电视信号主要由\_\_\_\_\_组成。  
(1) 图像信号、同步信号、消隐信号。  
(2) 图像信号、亮度信号、色度信号。  
(3) 图像信号、复合同步信号、复合消隐信号。  
(4) 图像信号、复合同步信号、复合色度信号。  
(A) (1) (B) (2)  
(C) (3) (D) (4)
3. 在 YUV 彩色空间中数字化后 Y:U:V 是\_\_\_\_\_。  
(1) 4:2:2 (2) 8:4:2  
(3) 8:2:4 (4) 8:4:4  
(A) (1) (B) (2)  
(C) (3) (D) (4)
4. 彩色全电视信号主要由\_\_\_\_\_组成。  
(1) 图像信号、亮度信号、色度信号、复合消隐信号。  
(2) 亮度信号、色度信号、复合同步信号、复合消隐信号。  
(3) 图像图信号、复合同步信号、消隐信号、亮度信号。  
(4) 亮度信号、同步信号、复合消隐信号、色度信号。  
(A) (1) (B) (2)  
(C) (3) (D) (4)
5. 数字视频编码的方式有哪些？  
(1) RGB 视频 (2) YUV 视频  
(3) Y/C (S) 视频 (4) 复合视频  
(A) (1) (B) (1) (2)  
(C) (1) (2) (3) (D) 全部

6. 在多媒体计算机中常用的图像输入设备是\_\_\_\_\_。

- (1) 数码照相机 (2) 彩色扫描仪  
(3) 视频信号数字化仪 (4) 彩色摄像机  
(A) (1) (B) (1) (2)  
(C) (1) (2) (3) (D) 全部

7. 下列数字视频中哪个质量最好?

- (1) 240×180 分辨率、24 位真彩色、15 帧/秒的帧率。  
(2) 320×240 分辨率、30 位真彩色、25 帧/秒的帧率。  
(3) 320×240 分辨率、30 位真彩色、30 帧/秒的帧率。  
(4) 640×480 分辨率、16 位真彩色、15 帧/秒的帧率。
- (A) (1) (B) (2)  
(C) (3) (D) (4)

8. 简述视频信息获取的流程, 并画出视频信息获取的流程框图。

9. 简述视频信号获取器的工作原理。

### 3.7 参考答案

1. (C)
2. (C)
3. (D)
4. (B)
5. (D)
6. (D)
7. (C)
8. (略)
9. (略)



## 第4章 多媒体数据压缩编码技术

### 本章要点

- 多媒体数据压缩编码的重要性和分类。
- 常用压缩编码算法的基本原理及实现技术，预测编码（DPCM、ADPCM）、变换编码（DCT变换）、统计编码（信息熵、Huffman编码、算术编码）。
- 静态图像压缩编码的国际标准 JPEG 的原理及实现技术，以及动态图像压缩编码国际标准 MPEG 的基本原理。

进入信息时代，人们越来越依靠计算机获取和利用信息。而数字化后的视频和音频等媒体信息具有媒体海量性，与当前硬件技术所能提供的计算机存储资源和网络带宽之间有很大差距。这样，就对多媒体信息的存储和传输造成了很大困难，成为阻碍人们有效获取和利用信息的一个瓶颈问题。从目前计算机的软硬件发展水平及发展趋势来看，可以断言：在将来很长的一段时期内，数字化的媒体信息数据以压缩形式存储和传播仍将是惟一的选择。

### 4.1 多媒体数据压缩的重要性和分类

#### 4.1.1 多媒体数据压缩的必要性

信息时代的重要特征是信息的数字化，数字化了的信息带来了“信息爆炸”。多媒体计算机系统技术是面向三维图形、立体声和彩色全屏幕运动画面的处理技术。数字计算机面临的是数值、文字、语言、音乐、图形、动画、静图像、电视视频图像等多种媒体承载的由模拟量转化成数字量信息的吞吐、存储和传输的问题。数字化了的视频和音频信号的数量之大是非常惊人的。下面列举几个未经压缩的数字化信息的例子：

(1) 一页印在 B5 (约 180mm×255mm) 纸上的文件，若以中等分辨率 (300dot/in 约 12 像素点/mm) 的扫描仪进行采样，其数据量约 6.61MB/页。一片 650MB 的 CD-ROM，可存 98 页。

(2) 双通道立体声激光唱盘 (CD-A)，采样频率为 44.1kHz，采样精度 16 位/样本，其一秒钟时间内的采样位数为  $44.1 \times 10^3 \times 16 \times 2 = 1.41 \text{ Mbit/s}$ 。一个 650MB 的 CD-ROM，可存约 1 小时的音乐。

(3) 数字音频磁带 (DAT)，采样频率 48kHz，采样精度 16 位/样本，一秒钟时间内采样位数为  $48 \times 10^3 \times 16 = 768 \text{ Kbit/s}$ ，一个 650MB 的 CD-ROM 可存储近 2 小时的节目。

(4) 数字电视图像：

1) SIF (source input format) 格式，NTSC 制、彩色、4:4:4 采样  
每帧数据量： $352 \times 240 \times 3 = 253 \text{ KB}$

每秒数据量（比特率）： $253 \times 30 = 7.59\text{MB/s}$

一片 CD-ROM 可存帧数： $650/0.253 = 2.569\text{K 帧/片}$

一片 CD-ROM 节目时间： $(650/7.59)/60 = 1.42 \text{ 分/片}$

2) ICCR (International Consultative Committee for Radio) 格式, PAL 制、4:4:4 采样

每帧数据量： $720 \times 576 \times 3 = 1.24\text{MB}$

每秒数据量： $1.24 \times 25 = 31.3\text{MB/S}$

一片 CD-ROM 可存帧数： $650/1.24 = 0.524\text{K 帧/片}$

一片 CD-ROM 可存节目时间： $650/31.1 = 20.9 \text{ 秒/片}$

我们再举一个陆地卫星 (LandSat-3) 的例子 (其水平、垂直分辨率分别为 2340 和 3240, 四波段、采样精度 7 位), 它的一幅图像的数据量为  $2340 \times 3240 \times 7 \times 4 = 212\text{Mbit}$ , 按每天 30 幅计, 每天数据量为  $212 \times 30 = 6.36\text{Gbit}$ , 每年的数据量高达  $2300\text{Gbit}$ 。

从以上列举的数据例子可以看出数字化信息的数据量是何等庞大, 这样大的数据量, 无疑给存储器的存储容量、通信干线的信道传输率以及计算机的速度都增加了极大的压力。这个问题是多媒体技术发展中的一个非常棘手的瓶颈问题。解决这一问题, 单纯用扩大存储器容量、增加通信干线的传输率的办法是不现实的。数据压缩技术是个行之有效的方法。通过数据压缩手段把信息数据量压下来, 以压缩形式存储和传输, 既紧缩节约了存储空间, 又提高了通信干线的传输效率, 同时也使计算机实时处理音频、视频信息, 以保证播放出高质量的视频、音频节目成为可能。

### 4.1.2 多媒体数据压缩的可能性

人们研究发现, 图像数据表示中存在着大量的冗余。通过去除那些冗余数据可以使原始图像数据极大地减少, 从而解决图像数据量巨大的问题。图像数据压缩技术就是研究如何利用图像数据的冗余性来减少图像数据量的方法。因此, 进行图像压缩研究的起点是研究图像数据的冗余性。

下面我们将介绍常见的一些图像数据冗余的情况。

#### 1. 空间冗余

这是静态图像存在的最主要的一种数据冗余。一幅图像记录了画面上可见景物的颜色。同一景物表面上各采样点的颜色之间往往存在着空间连贯性, 但是基于离散像素采样来表示物体颜色的方式通常没有利用景物表面颜色的这种空间连贯性, 从而产生了空间冗余。我们可以通过改变物体表面颜色的像素存储方式来利用空间连贯性, 达到减少数据量的目的。例如: 在静态图像中有一块表面颜色均匀的区域, 在此区域中所有点的光强和色彩以及饱和度都是相同的, 因此数据有很大的空间冗余。

#### 2. 时间冗余

这是序列图像 (电视图像、运动图像) 表示中经常包含的冗余。序列图像一般为位于一时间轴区间内的一组连续画面, 其中的相邻帧往往包含相同的背景和移动物体, 只不过移动物体所在的空间位置略有不同, 所以后一帧的数据与前一帧的数据有许多共同的地方, 这种共同性是由于相邻帧记录了相邻时刻的同一场景画面, 所以称为时间冗余。

#### 3. 结构冗余

有些图像的纹理区, 图像的像素值存在着明显的分布模式。例如, 方格状的地板图案等。

我们称此为结构冗余。已知分布模式，可以通过某一过程生成图像。

#### 4. 知识冗余

有些图像的理解与某些基础知识有相当大的相关性。例如：人脸的图像有固定的结构，比如说嘴的上方有鼻子，鼻子的上方有眼睛，鼻子位于脸的中线上等等。这类规律性的结构可由先验知识和背景知识得到称此类冗余为知识冗余。根据已有的知识，对某些图像中所包含的物体，可以构造其基本模型，并创建对应各种特征的图像库，进而图像的存储只需要保存一些特征参数，从而可以大大减少数据量。知识冗余是模型编码主要利用的特性。

#### 5. 视觉冗余

事实表明，人类的视觉系统对图像场的敏感性是非均匀和非线性的。然而，在记录原始的图像数据时，通常假定视觉系统是线性的和均匀的，对视觉敏感和不敏感的部分同等对待，从而产生了比理想编码（即把视觉敏感和不敏感的部分区分开来编码）更多的数据，这就是视觉冗余。通过对人类视觉进行的大量实验，发现了以下的视觉非均匀特性。

(1) 视觉系统对图像的亮度和色彩度的敏感性相差很大。当把 RGB 颜色空间转化成 NTSC 制的 YIQ 坐标系后，经实验发现，视觉系统对亮度 Y 的敏感度远远高于对色彩度 (I 和 Q) 的敏感度。因此对色彩度 (I 和 Q) 允许的误差可大于对亮度 Y 所允许的误差。

(2) 随着亮度的增加，视觉系统对量化误差的敏感度降低。这是由于人眼的辨别能力与物体周围的背景亮度成反比。由此说明：在高亮度区，灰度值的量化可以更粗糙些。

(3) 人眼的视觉系统把图像的边缘和非边缘区域分开来处理。这是将图像分成非边缘区域和边缘区域分别进行编码的主要依据。这里的边缘是指灰度值发生剧烈变化的地方，而非边缘区域是指除边缘之外的图像其他任何部分。

(4) 人类的视觉系统总是把视网膜上的图像分解成若干个空间有向的频率通道后再进一步处理。在编码时，若把图像分解成符合这一视觉内在特性的频率通道，则可能获得较大的压缩比。以后我们提到的小波编码就是在一定程度上利用了这一特性。

#### 6. 图像区域的相同性冗余

它是指在图像中的两个或多个区域所对应的所有像素值相同或相近而产生的数据重复性存储，这就是图像区域的相似性冗余。在以上情况下，记录了一个区域中各像素的颜色值，则与其相同或相近的其他区域就不再需记录其中各像素的值。向量量化 (vector quantization) 方法就是针对这种冗余性的图像压缩编码方法。

#### 7. 纹理的统计冗余

有些图像纹理尽管不严格服从某一分布规律，但是它在统计的意义上服从该规律。利用这种性质也可以减少表示图像的数据量，所以我们称之为纹理的统计冗余。

随着对人类视觉系统和图像模型的进一步研究，人们可能会发现更多的冗余性，使图像数据压缩编码的可能性越来越大，从而推动图像压缩技术的进一步发展。

### 4.1.3 多媒体数据压缩方法的分类

多媒体数据压缩方法根据不同的依据可产生不同的分类。第一种，根据质量有无损失可分为有损失编码和无损失编码。第二种，按照其作用域在空间域或频率域上分为空间方法、变换方法和混合方法。第三种，根据是否自适应分为自适应性编码和非适应性编码。一般来说，每一个编码方法都有其相应的自适应方法。

下面我们介绍一种多媒体数据依据压缩算法进行分类的方法，并给出简单的说明解释。  
如图 4-1 所示。

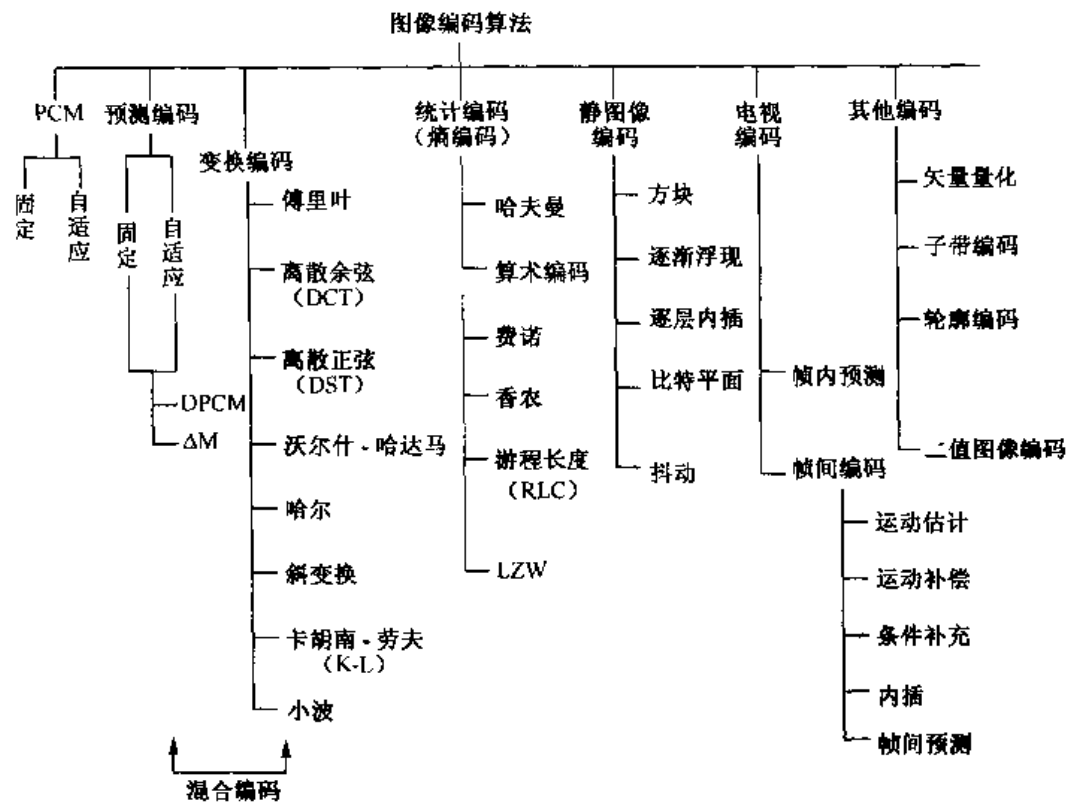


图 4-1 多媒体数据编码分类

### 1. 脉冲代码调制 PCM (Pulse Code Modulation)

它实际上是连续模拟信号的数字采样表示。通常使用 Nyquist 采样速率。若量化器为  $N$  级， $N=2^b$ ，则每一个采样用  $b$  位的二进制代码表示。在信号的量化中，每一色彩分量一般用 8 位表示。PCM 编码器和解码器位于一个图像编码系统的起点和终点。它们实际上分别是 A/D 转换器和 D/A 转换器，我们以下讨论的编码方法都是在多媒体数据模拟信号经过 PCM 编码后再进行的压缩编码方法。

### 2. 预测编码

编码器记录与传输的不是样本的真实值，而是它与预测值的差。这一方法称为 DPCM (Differential Pulse Code Modulation) 方法。预测值由欲编码图像信号的过去信息决定。通常采用线性预测，比例系数由其统计特性估计得到。预测不仅可以在相邻像素值之间进行，而且可以在行与行之间进行。由于空间相关性，真实值与预测值的差值变化范围远远小于真实值的变化范围，因而可以采用较少的位数来表示。另外，若利用人的视觉特性对差值进行非均匀量化，则会获得更高的压缩比。

### 3. 变换编码

其主要思想是利用图像块内像素值之间的相关性，把图像变换到一组新的基上，使得能量集中到少数几个变换系数上，通过存储这些系数而达到压缩的目的。在变换编码中，由于对整幅图像进行变换的计算量太大，所以一般把原始图像分成许多个矩形区域子图像独立进行变换。常用的变换有 KLT (Karhunen-Loeve Transform)，DCT (Discrete Cosine Transform)，

基础是现代统计学和控制论。由于数字技术的飞速发展,数字信号处理技术不时渗透到这些领域,在这些理论与技术的基础上形成了一个专门用作压缩冗余数据的预测编码技术。预测编码主要是减少了数据在时间和空间上的相关性,因而对于时间序列数据有着广泛的应用价值。在数字通信系统中,例如语音的分析与合成,图像的编码与解码,预测编码已得到了广泛的实际应用。

预测编码是根据某一模型利用以往的样本值对于新样本值进行预测,然后将样本的实际值与其预测值相减得到一个误差值,对于这一误差值进行编码。如果模型足够好且样本序列在时间上相关性较强,那么误差信号的幅度将远远小于原始信号,从而可以用较少的电平类对其差值量化得到较大的数据压缩结果。

如果能精确预测数据源输出端作为时间函数使用的样本值的话,就不存在关于数据源的不确定性,因而也就不存在要传输的信息。换句话说,如果我们能得到一个数学模型完全代表数据源,那么在接收端就能依据这一数学模型精确地产生出这些数据。然而没有一个实际的系统能找到其完整的数据模型,我们能找到的最好的预测器是以某种最小化的误差对下一个采样进行预测的预测器。

通常预测器的设计不是利用数据源的实际数学模型,因为数据源的实际数学模型是非常复杂的,而且是时变的,以至于对实际的应用根本无法求得。相反,利用输出产生的任何影响并不直接涉及到数据源,所以预测器可以独立进行工作,而且不会影响数据源样本。利用样本的预测器可以以这些样本的线性或非线性函数为基础,大部分的预测都采用了线性预测函数。科尔莫戈罗夫(1941年)和维纳(1942年)进行了关于线性预测的开创性工作,他们以最小均方量化误差为准则,从而建立了以最小均方预测误差为最优预测,和以最小量化误差作为最优量化基础的预测原理。但同时,人们也注意到了人的主观评价准则,既依据人的主观评价设计预测器,这样也导致了人们对信源的主观评价的进一步研究,寻求适合人的主观评价的预测算法。

### 4.2.1 预测编码的基本原理

模拟量经 A/D 变换,得到二进制码的过程,就是著名的脉冲编码调制 PCM(Pulse Code Modulation)编码过程,也称 PCM 编码。

对于图像信号,经 A/D 变换后,为避免假轮廓出现,黑白单色图像灰度(亮度)量化级是 256 级分层,8 位 PCM 编码。红、绿、兰或 Y、U、V 彩色图像信号也分别以 8 位 PCM 编码。

PCM 编码是 8 位等长二进制码,其编码率(每像素所需的位数)不够小,比如,对于 256 级灰度的黑白图像,每像素需 8 位(8bits/Pel);对于彩色图像,每像素需 24 位(24bits/Pel)。所以直接以 PCM 编码存储或传送数字图像,其总数据量还是太庞大,无法实现,因此需采用更高压缩比的压缩编码方法。预测编码方法是一种较实用且被广泛采用的一种压缩编码方法。预测编码方法原理是从相邻像素之间有强的相关性特点考虑的。比如当前像素的灰度或颜色信号在数值上与其相邻像素总是比较接近,除非处于边界状态。那么,当前像素的灰度或颜色信号的数值可用前面已出现的像素的值进行预测(估计),得到一个预测值(估计值),将实际值与预测值求差,对这个差值信号进行编码、传送,这种编码方法称为预测编码方法。

预测编码方法分线性预测和非线性预测编码方法。线性预测编码方法也称差值脉冲编码

调制法，简称 DPCM（Differention Pulse Code Modulation）。预测编码方法在图像数据压缩和语音信号的数据压缩中都得到广泛的应用和研究。

1. DPCM 的基本原理

一幅二维静止图像，设空间坐标  $(i, j)$  像素点的实际灰度为  $f(i, j)$ ， $\hat{f}(i, j)$  是根据以前已出现的像素点的灰度对该点的预测灰度，也称预测值或估计值，计算预测值的像素，可以是同一扫描行的前几个像素，或者是前几行上的像素，甚至是前几帧的邻近像素。实际值和预测值之间的差值，以下式表示：

$$e(i, j) = f(i, j) - \hat{f}(i, j) \tag{4-1}$$

将此差值定义为预测误差，由于图像像素之间有极强的相关性，所以这个预测误差是很小的。编码时，不是对像素点的实际灰度  $f(i, j)$  进行编码，而是对预测误差信号  $e(i, j)$  进行量化、编码、发送，由此而得名为差值脉冲编码调制法，简写 DPCM。

DPCM 编、解码系统原理图如图 4-2 所示。系统包括发送、接收和信道传输三个部分。发送端由编码器、量化器、预测器和加/减法器组成；接收端包括解码器和预测器等；信道传送以虚线表示。由图可见 DPCM 系统具有结构简单，容易用硬件实现（接收端的预测器和发送端的预测器完全相同）的优点。图中输入信号  $f(i, j)$  是坐标为  $(i, j)$  像素点的实际灰度值， $\hat{f}(i, j)$  是由已出现先前相邻像素点的灰度值对该像素点的预测灰度值。 $e(i, j)$  是预测误差。假如发送端不带量化器，直接对预测误差  $e(i, j)$  进行编码、传送，接收端可以无误差地恢复  $f(i, j)$ 。这是可逆的无失真的 DPCM 编码，是信息保持编码；但是，如果包含量化器，这时编码器对  $e'(i, j)$  编码，量化器导致了不可逆的信息损失，这时接收端经解码恢复出的灰度信号不是真正的  $f(i, j)$ ，而是以  $f'(i, j)$  表示这时的输出。可见引入量化器会引起一定程度的信息损失，使图像质量受损。但是，为了压缩位数，利用人眼的视觉特性，对图像信息丢失不易觉察的特点，带有量化器有失真的 DPCM 编码系统还是普遍被采用。

2. 最佳线性预测

图 4-3 给出像素  $(i, j)$  的预测域图，图中示出  $f(i, j)$  像素的三个相邻像素， $\hat{f}(i, j)$  由先前（同行一点，上一行两点）三点预测，定义为：

$$\hat{f}(i, j) = a_1 f(i, j-1) + a_2 f(i-1, j-1) + a_3 f(i-1, j) \tag{4-2}$$

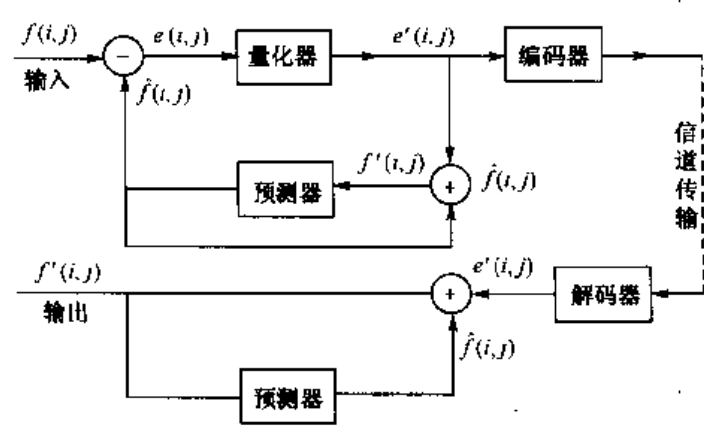


图 4-2 DPCM 编、解码原理图

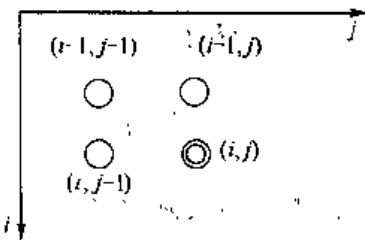


图 4-3 预测域

构成三阶预测器。其中  $a_1, a_2, a_3$  称预测系数，都是待定参数。如果预测器中预测系数是固定不变的常数，称之为线性预测。

预测误差：

$$\begin{aligned} e(i, j) &= f(i, j) - \hat{f}(i, j) \\ &= f(i, j) - [a_1 f(i, j-1) + a_2 f(i-1, j-1) + a_3 f(i-1, j)] \end{aligned}$$

图 4-4 给出一个典型的三阶线性预测 DPCM 框图。其中虚线框是三阶线性预测器，接收端具有与发送端完全相同的三阶线性预测器。线性预测器中  $a_1, a_2, a_3$  是待定参数，当  $a_1, a_2, a_3$  满足使预测误差最小且保持固定不变时，便构成最佳线性预测器。

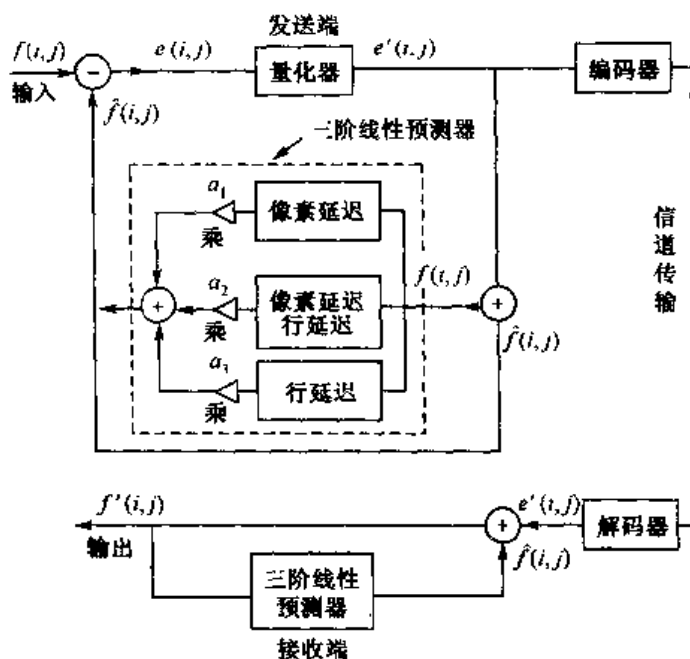


图 4-4 三阶 DPCM 线性预测框图

现在我们以图 4-4 这个三阶线性预测器，应用均方误差最小准则，求出预测系数  $a_1, a_2, a_3$ ，以获得  $f(i, j)$  的最佳线性预测值  $\hat{f}(i, j)$ 。

均方误差的表达式为

$$\begin{aligned} \bar{e}^2 &= E\{[e(i, j)]^2\} \\ &= E\{[f(i, j) - \hat{f}(i, j)]^2\} \\ &= E\{[f(i, j) - a_1 f(i-1, j) - a_2 f(i-1, j-1) - a_3 f(i, j-1)]^2\} \end{aligned} \quad (4-3)$$

将预测值与实际值之间的均方误差  $\bar{e}^2$  对  $a_1, a_2, a_3$  求偏导，令

$$\begin{cases} \frac{\partial \bar{e}^2}{\partial a_1} = 0 \\ \frac{\partial \bar{e}^2}{\partial a_2} = 0 \\ \frac{\partial \bar{e}^2}{\partial a_3} = 0 \end{cases} \quad (4-4)$$

解方程，得  $a_1$ ， $a_2$ ， $a_3$ ，即为最佳线性预测系数。

可以证明，假定图像信号符合平稳的马尔可夫过程的要求，那么，可以直接用相关系数来确定预测系数。

## 4.2.2 自适应预测编码

前面叙述的 DPCM 系统是预测系数和量化器参数一次设计好后，整幅图都用这套参数，不再改变。但是在图像平坦区和边缘处要求量化器的输出差别很大，否则会导致图像出现令人讨厌的噪声，下面将简要介绍一下关于自适应 DPCM 系统，即 ADPCM (Adaptive DPCM) 系统。自适应技术的概念是预测器的预测系数和量化器的量化参数，能够根据图像的局部区域分布特点而自动调整。实践证明 ADPCM 编、解码系统与 DPCM 编、解码系统相比，不仅能改善恢复图像的评测质量和视觉效果，同时还能进一步压缩数据。

ADPCM 系统包括自适应预测，即预测系数的自适应调整；自适应量化，即量化器参数的自适应调整两部分内容。

### 1. 自适应预测

由公式 (4-2) 可知一个三阶预测器的预测值计算公式为

$$\hat{f}(i, j) = a_1 f(i, j-1) + a_2 f(i-1, j-1) + a_3 f(i-1, j)$$

现在增加一个可变参数“ $m$ ”，得

$$\hat{f}(i, j) = m[a_1 f(i, j-1) + a_2 f(i-1, j-1) + a_3 f(i-1, j)] \quad (4-5)$$

式中  $m$  是一个自适应参数， $m$  的取值据量化误差的大小自适应调整。

设量化器最大输出为  $e_{\max}$ ，最小输出为  $e_{\min}$ ，某一个预测误差的量化输出为  $e'$ ，

当  $e_{\min} < |e'| < e_{\max}$   $m$  不变

$|e'| = e_{\max}$   $m$  自动增大

$|e'| = e_{\min}$   $m$  自动减小

$m$  自动增大，使  $\hat{f}(i, j)$  随之增大，预测误差减小，使斜率过载尽快收敛； $m$  自动减小，使  $\hat{f}(i, j)$  随之减小，预测误差加大，使量化器输出不致正负跳变，减轻颗粒噪声。大多数情况  $|e'|$  处于  $e_{\min}$  和  $e_{\max}$  之间， $m$  取常数不变。

### 2. 自适应量化

自适应量化是根据图像局部区域的特点，自适应地修改和调整量化器的参数，包括量化器输出的动态范围，量化器判决电平（量化器步长）等，下面介绍一种利用视觉阈值曲线导出自适应量化特性的方法，这种情况下的量化属于非线性量化。实际上是在量化器分层确定



后,当预测误差值小时,将量化器的输出动态范围减小,量化器步长减小;当预测误差大时,将量化器的输出范围扩大,量化器步长扩大。参数改变的原则,是量化误差低于该误差下的视觉阈值,将误差掩盖。

自适应量化的具体实现方法是,先定义一视觉掩盖函数  $M$ ,

$$M = \max\{|f_1 - f|, |f_2 - f|, |f_3 - f|, |f_4 - f|\} \quad (4-6)$$

其中

$$\begin{aligned} |e_1| &= |f_1 - f| \\ |e_2| &= |f_2 - f| \\ |e_3| &= |f_3 - f| \\ |e_4| &= |f_4 - f| \end{aligned}$$

$f_1, f_2, f_3, f_4$  是  $f$  点相邻像素点的灰度值,如图 4-5 所示。

由公式 (4-6) 定义的掩盖函数的含义是,当四个差值  $e_1, e_2, e_3, e_4$  中有一个较大数值,那么对预测  $f$  所形成的量化误差构成“掩盖效应”,即掩盖量化噪声,使人眼难以察觉。

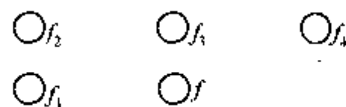


图 4-5  $f$  像素的相邻像素

设量化分层级数为 16,确定以下四种情况下的量化输出电平值。

(1) 当视觉掩盖函数  $M < 20$  时,只取  $|e| = 0$  两侧的 16 个量化分层,量化器步长较细。

(2) 当视觉掩盖函数  $20 \leq M < 36$  时,  $M=20$ ,可见度阈值约 3.5,那么 3.5 以内的量化误差可掩盖,这时略去  $|e| = “5”$ ,补“59”,增加较大步距的量化输出。

(3) 当  $36 \leq M < 72$  时,  $M=36$  的可见度阈值约 5.5,略去“2”,“8”,取回“5”,再补充一个较大的量化输出。

(4) 当  $M > 72$  时,  $M=72$  的可见度阈值约 7.5,略去“2”、“5”、“8”和“20”,量化输出可扩大到 100。

从以上过程看到,根据  $M$  的大小,改变量化器的步长,从而实现了自适应量化。

## 4.3 变换编码

### 4.3.1 变换编码的基本原理综述

变换编码不是直接对空域图像信号编码,而是首先将空域图像信号映射变换到另一个正交矢量空间(变换域或频域),产生一批变换系数,然后对这些变换系数进行编码处理。图 4-6 示出一个变换编解码进行过程的示意图。在发送端将原始图像分割成 1 到  $n$  个子图像块,每个子图像块送入正交变换器作正交变换,变换器输出变换系数经滤波、量化、编码后送信道传输到达接收端,接收端作解码、逆变换、综合拼接,恢复出空域图像。

数字图像信号经过正交变换为什么能够压缩数据量呢?先让我们看一个最简单的时域三角函数  $y(t) = A \sin 2\pi ft$  的例子,当  $t$  从  $-\infty$  到  $+\infty$  改变时,  $y(t)$  是一个正弦波。假如将其变换到频域表示,只需幅值  $A$  和频率  $f$  两个参数就足够了,可见  $y(t)$  在时域描述,数据之间的相关性大,数据冗余度大;而转换到频域描述,数据相关性大大减少,数据冗余量减少,参数独立,数据量减少。

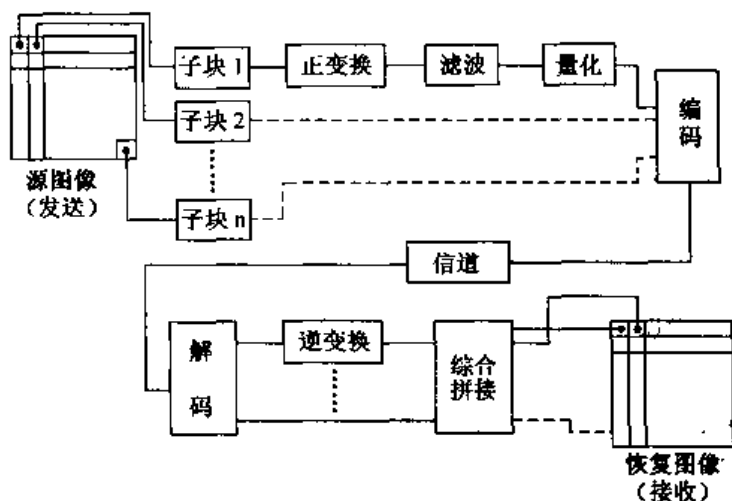


图 4-6 变换编、解码过程示意图

现在举一个例子，设有两个相邻的数据样本  $x_1$  与  $x_2$ ，每样本采用 3 位编码，因此各有  $2^3=8$  个幅度等级。而两个样本的联合事件，共有  $8 \times 8=64$  种可能性，可用图 4-7 的二维平面坐标表示。其中  $x_1$  轴与  $x_2$  轴分别表示相邻两样本可能的幅度等级。对于慢变信号，相邻两样本  $x_1$  与  $x_2$  同时出现相近幅度等级的可能性较大，因此，图 4-7a 阴影区内  $45^\circ$  斜线附近的联合事件出现概率也就较大，不妨将此阴影区之边界称为相关圈。信源的相关性愈强，则相关圈愈加扁长。或者形象地说， $x_1$  与  $x_2$  呈现“水涨船高”的紧密关联特性。为了要对圈内各点的位置进行编码，就要对两个差不多大的坐标值分别进行编码。当相关性愈弱时，此相关圈就愈显方圆形状，说明  $x_1$  处于某一幅度等级时， $x_2$  可能出现在不相同的任意幅度等级上。

现在若对该数据对进行正交变换，从几何上相当于坐标系旋转  $45^\circ$ ，变成  $y_1, y_2$  坐标系，如图 4-7b 所示。那么此时该相关圈正好处在  $y_1$  坐标轴上下，且该圈愈扁长，其在  $y_1$  上的投影就愈大，而在  $y_2$  上的投影就愈小。因而从  $y_1, y_2$  坐标来看，任凭  $y_1$  在较大范围内变化，而  $y_2$  却巍然“不动”或“只有微动”。这就意味着变量  $y_1$  和  $y_2$  之间在统计更加相互独立。因此，通过这种坐标系旋转变换，就能得到一组去除掉大部分甚至全部统计相关性的另一种输出样本。

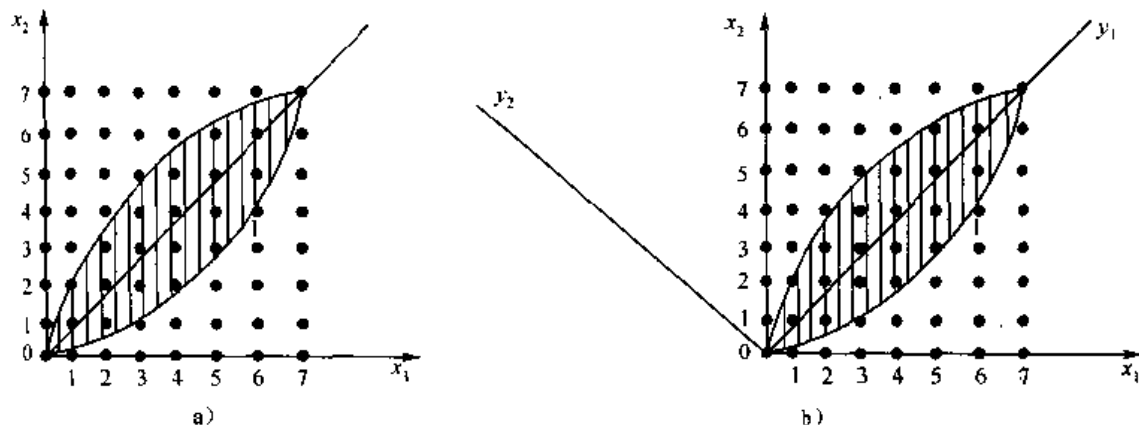


图 4-7 正交变换的几何意义

a) 子图像在阴影区中的概率大小 b) 坐标系旋转后的情况

变换编码技术已有近 30 年的历史，技术上比较成熟，理论也比较完备，广泛应用于各

种图像数据压缩, 诸如单色图像、彩色图像、静止图像、运动图像以及多媒体计算机技术中的电视帧内图像压缩和帧间图像压缩等。

正交变换的种类很多, 如傅立叶 (Fouries) 变换、沃尔什 (Walsh) 变换、哈尔 (Haar) 变换、斜 (slant) 变换、余弦变换、正弦变换、K-L (Karhunen - Loeve) 变换等。

### 4.3.2 离散余弦变换——DCT 变换

余弦变换是傅里叶变换的一种特殊情况。在傅里叶级数展开式中, 如果被展开的函数是实偶函数, 那么, 其傅里叶级数中只包含余弦项, 再将其离散化由此可导出余弦变换, 或称之为离散余弦变换 DCT (Discrete Cosine Transform)。

离散余弦变换在数字图像数据压缩编码技术中可与最佳变换 K-L 变换媲美, 因为 DCT 与 K-L 变换压缩性能和误差很接近, 而 DCT 计算复杂度适中, 又具有可分离特性, 还有快速算法等特点, 所以近年来在图像数据压缩中, 采用离散余弦变换编码的方案很多, 特别是 20 世纪 90 年代迅速崛起的计算机多媒体技术中, JPEG、MPEG、H.261 等压缩标准, 都用到离散余弦变换编码进行数据压缩。

#### 1. 一维离散余弦变换

设一维离散函数  $f(x)$ ,  $x = 0, 1, \dots, N-1$ , 把  $f(x)$  扩展成为偶函数的方法有两种, 以  $N=4$  为例, 可得出如图 4-8 和图 4-9 所示的两种情况。图 4-8 为偶对称, 图 4-9 为奇对称, 从而有偶离散余弦变换 (EDCT) 和奇离散余弦变换 (ODCT)。

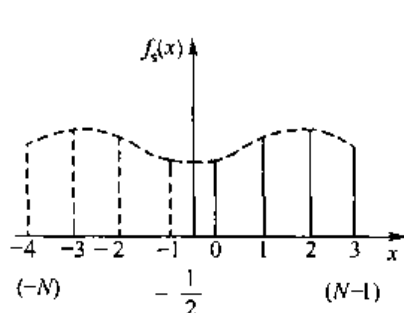


图 4-8 偶对称

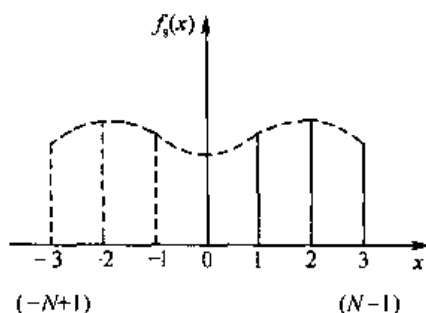


图 4-9 奇对称

由上示二图看出, 对于偶对称扩展, 对称轴在  $x = -\frac{1}{2}$  处。

$$f_s(x) = \begin{cases} f(x) & \text{当 } 0 \leq x \leq N-1 \\ f(-x-1) & \text{当 } -N \leq x \leq -1 \end{cases} \quad (4-7)$$

采样点数增到  $2N$ 。

奇对称扩展, 对称轴在  $x = 0$  处。

$$f_s(x) = \begin{cases} f(x) & \text{当 } 0 \leq x \leq N-1 \\ f(-x) & \text{当 } -N+1 \leq x \leq -1 \end{cases} \quad (4-8)$$

采样点数增到  $2N-1$ 。

由离散傅里叶变换定义出发, 对公式 (4-7) 作傅里叶变换, 以  $F_s(u)$  表示, 则得

$$\begin{aligned}
F_s(u) &= \frac{1}{2N} \sum_{x=-N}^{N-1} f_s(x) e^{-\frac{j2\pi}{2N}u(x+\frac{1}{2})} \\
&= \frac{1}{2N} \sum_{x=-N}^{N-1} f_s(x) e^{-\frac{j(2x+1)}{2N}u\pi} \\
&= \frac{1}{2N} \sum_{x=-N}^{N-1} f_s(x) \cos\left(\frac{2x+1}{2N}u\pi\right) \\
&= \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cos\left(\frac{2x+1}{2N}u\pi\right)
\end{aligned} \tag{4-9}$$

式中,  $u = -N, \dots, N-1$

$$\text{当 } u=0, F_s(0) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)$$

$$\text{当 } u=-N, F_s(-N) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cos\left(-x\pi - \frac{\pi}{2}\right) = 0$$

当  $u = \pm 1, \pm 2, \dots, \pm(N-1)$ ,  $F_s(u) = F_s(-u)$ , 且

$$F_s(u) + F_s(-u) = 2 \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cos\left(\frac{2x+1}{2N}u\pi\right)$$

考虑正变换公式与逆变换公式的对称性, 令

$$u=0, \quad C(0) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N-1} f(x) \tag{4-10}$$

$$u=1, 2, \dots, N-1, \quad C(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos\left(\frac{2x+1}{2N}u\pi\right) \tag{4-11}$$

式中,

$$u=0, \quad g(0, x) = \sqrt{\frac{1}{N}} \tag{4-12}$$

$$u=1, 2, \dots, N-1, \quad g(u, x) = \sqrt{\frac{2}{N}} \cos\left(\frac{2x+1}{2N}u\pi\right) \tag{4-13}$$

定义式(4-10)和式(4-11)为离散偶余弦正变换公式; 式(4-12)和式(4-13)为离散偶余弦变换核公式。

离散偶余弦逆变换公式为

$$f(x) = \sqrt{\frac{1}{N}} C(0) + \sqrt{\frac{2}{N}} \sum_{u=1}^{N-1} C(u) \cos\left(\frac{2x+1}{2N} u\pi\right) \quad (4-14)$$

式中  $x = 0, 1, \dots, N-1$

将公式 (4-10) 和公式 (4-11) 合并、化简, 可得到一维离散偶余弦正变换公式, 即

$$C(u) = E(u) \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos\left(\frac{2x+1}{2N} u\pi\right) \quad (4-15)$$

式中  $u = 0, 1, \dots, N-1$

当  $u = 0$  时,  $E(u) = 1/\sqrt{2}$

当  $u = 1, 2, \dots, N-1$  时,  $E(u) = 1$

## 2. 二维离散偶余弦变换

设空域变量取值范围为

$x = 0, 1, \dots, N-1$

$y = 0, 1, \dots, N-1$

频域变量取值范围为

$u = 0, 1, \dots, N-1$

$v = 0, 1, \dots, N-1$ , 那么,

二维离散偶余弦正变换公式为

$$C(u, v) = E(u)E(v) \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cdot \cos\left(\frac{2x+1}{2N} u\pi\right) \cdot \cos\left(\frac{2y+1}{2N} v\pi\right) \quad (4-16)$$

式中  $u, v = 0, 1, \dots, N-1$

$E(u), E(v) = 1/\sqrt{2}$  当  $u = 0, v = 0$  时

$E(u), E(v) = 1$  当  $u = 1, 2, \dots, N-1; v = 1, 2, \dots, N-1$  时, 二维离散偶余弦逆变换公式

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} E(u)E(v)C(u, v) \cdot \cos\left(\frac{2x+1}{2N} u\pi\right) \cdot \cos\left(\frac{2y+1}{2N} v\pi\right) \quad (4-17)$$

式中  $x, y = 0, 1, \dots, N-1$

$E(u), E(v) = 1/\sqrt{2}$  当  $u = 0, v = 0$  时

$E(u), E(v) = 1$  当  $u = 1, 2, \dots, N-1; v = 1, 2, \dots, N-1$  时

二维离散余弦变换核具有可分离特性, 所以, 其正变换和逆变换均可将二维变换分解成一系列一维变换 (行、列) 进行计算。

## 3. 借助 FFT 实现离散余弦变换

由公式 (4-10) 和公式 (4-11) 一维离散偶余弦正变换公式略加变换, 即

$$\text{当 } u=0, \quad C(u) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N-1} f(x)$$

当  $u=1, 2, \dots, N-1$

$$\begin{aligned} C(u) &= \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos\left(\frac{2x+1}{2N} u\pi\right) \\ &= \sqrt{\frac{2}{N}} R_e \left\{ \left[ e^{-j\frac{u\pi}{2N}} \right] \cdot \left[ \sum_{x=0}^{2N-1} f(x) e^{-j\frac{2\pi}{2N} ux} \right] \right\} \end{aligned} \quad (4-18)$$

式中,  $\left[ \sum_{x=0}^{2N-1} f(x) e^{-j\frac{2\pi}{2N} ux} \right]$  可用 FFT 算法计算, 其结果乘以  $e^{-j\frac{u\pi}{2N}}$ , 取实部即可得到离

散余弦变换结果。计算 FFT 时,  $x=0, 1, \dots, 2N-1$  求和, 但实际上, 在  $x=N, N+1, \dots, 2N+1$  范围内,  $f(x)$  均得零, 故仍然计算  $N$  个点。

#### 4. 二维快速离散余弦变换

二维快速离散余弦变换算法, 是直接对二维图像数据  $M \times N = 2^y \times 2^b$  逐层对半分块, 并重新排列数据, 直至被分割的子块尺寸为  $1 \times 1$  为止。这种算法既不是将二维分离成行、列, 再进行一系列的一维变换算法, 也不是借助于 FFT, 再取实部的算法。二维快速余弦变换, 只需做实数乘法和加法, 对于  $x$  方向取样点数为  $M$ ,  $y$  方向采样点数为  $N$  的  $f(x, y)$  图像数据块, 其快速余弦变换的实数乘法次数为  $3/8MN \cdot \log_2(MN)$ 。为了公式推导简化, 把采样点数  $MN$  和常数 4 都放在正变换式中。

正变换 (DCT):

$$C(u, v) = \frac{4}{MN} E(u)E(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot \cos\left(\frac{2x+1}{2M} u\pi\right) \cdot \cos\left(\frac{2y+1}{2N} v\pi\right) \quad (4-19)$$

式中,  $u=0, 1, \dots, M-1$ ;  $v=0, 1, \dots, N-1$

其中,  $E(u), E(v) = \begin{cases} 1/\sqrt{2} & \text{当 } u=v=0 \text{ 时} \\ 1 & \text{其余} \end{cases}$

逆变换 (IDCT):

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} E(u)E(v)C(u, v) \cdot \cos\left(\frac{2x+1}{2M} u\pi\right) \cdot \cos\left(\frac{2y+1}{2N} v\pi\right) \quad (4-20)$$

式中  $x=0, 1, \dots, M-1$ ;  $y=0, 1, \dots, N-1$

其中,  $E(u), E(v) = \begin{cases} 1/\sqrt{2} & \text{当 } u=v=0 \text{ 时} \\ 1 & \text{其余} \end{cases}$

4.4 统计编码

数据压缩技术的理论基础是信息论。根据信息论的原理，可以找到最佳数据压缩编码方法，数据压缩的理论极限是信息熵。如果要求在编码过程中不丢失信息量，即要求保存信息熵，这种信息保持编码又叫做熵保存编码，或者叫熵编码。熵编码是无失真数据压缩，用这种编码结果经解码后可无失真地恢复出原图像。当考虑到人眼对失真不易觉察的生理特征时，有些图像编码不严格要求熵保存，信息可允许部分损失以换取高的数据压缩比，这种编码是有失真数据压缩，通常运动图像的数据压缩是有失真编码，这就是著名的香农(Shannon)率失真理论，即信息编码率与允许的失真关系的理论。本节只讨论无失真熵保存编码方法。

4.4.1 统计编码原理——信息量和信息熵

图像的概率分布、信息量和信息熵之间有什么关系？在图像编码压缩研究理论中，为什么要引入信息论中“熵”值的概念，有什么重要意义？这是我们下面需要说明的问题。

以一个信源编码器模型来进行说明，见图 4-10。

其中，

$X$  是消息集，由几个信号单元  $x_j$  构成 ( $j=1, 2, \dots, n$ )。

$Z$  是输出集，由几个码字  $z_j$  构成 ( $j=1, 2, \dots, n$ )， $z_j$  与  $x_j$  一一对应。

$A_m$  是符号集，有  $m$  个码元  $a_i$  构成 ( $i=1, 2, \dots, m$ )，符号集中的码元组成输出码字。

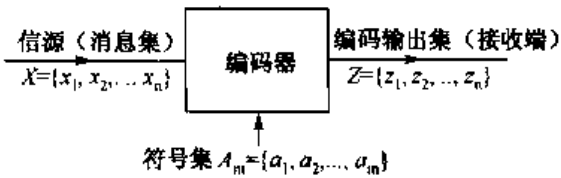


图 4-10 编码器模型图

当信源发出某个随机事件(消息)  $x_j$  后，接收端收到一个相应的码字  $z_j$ ，从数量上说，所收到的码字中包含多大的信息量，或者说多少有用的信息呢？

信息是用不确定性的量度定义的。一个消息的可能性愈小，其信息愈多；而消息的可能性愈大，则其信息愈少。在数学上讲，所传输的消息是其出现概率的单调下降函数。所谓信息量是指从  $N$  个相等可能事件中选出一个事件所需要的信息度量或含量，也就是在辨识  $N$  个事件中特定的一个事件的过程中所需要提问“是或否”的最少次数。例如，要从 64 个数中选定某一个数，可以先提问“是否大于 32”，不论回答是或否都消去了半数的可能事件，这样继续问下去，只要提问 6 次这类问题，就能从 64 个数中选定某一个数。这是因为每提问一次都会得到 1 位的信息量，因此在 64 个数中选定某一个数所需要的信息量是：

$$\log_2 64=6(\text{bits})$$

设从  $N$  个数中选定一个数  $x$  的概率为  $p(x)$ ，假定选定任意一个数的概率都相等，即  $P(x)=\frac{1}{N}$ ，则信息量为：

$$I(x_j)=\log_2 N=-\log_2 \frac{1}{N} = -\log_2 P(x)$$

信息论定义了一种度量信息量的方法，

$$I(x_j)=-\log_2 P(x_j) \quad j=1,2,\dots,n$$

其中,  $p(x_j)$  是信源  $X$  发出  $x_j$  先验概率。 $I(x_j)$  的含义是, 信源  $X$  发出  $x_j$  这个消息 (随机事件) 后, 接收端收到信息量的量度, 或者说接收端可能收到信源发出的是那一个随机事件的不确定性。

显然, 当随机事件  $x_j$  发生的先验概率  $p(x_j)$  大时, 由上式计算出的  $I(x_j)$  小, 那么这个事件发生的可能性大, 不确定性小, 事件一旦发生后提供的信息量也少。必然事件的  $p(x_j)$  等于 1,  $I(x_j)$  等于 0, 所以必然事件的消息报导不含任何信息量, 比如“太阳从东方升起”这个人所共知的事实, 毫无信息价值可言; 但是一件人们都没有估计到的事件 ( $p(x_j)$  极小) 一旦发生后,  $I(x_j)$  大, 包含的信息量很大, 即所谓爆炸性新闻。所以随机事件的先验概率, 与事件发生后所产生的信息量有密切关系。 $I(x_j)$  称  $x_j$  发生后的自信息量, 它也是一个随机变量。

信源  $X$  发出的  $x_j (j=1, 2, \dots, n)$  共  $n$  个随机事件的自信息统计平均 (求数学期望), 即

$$\begin{aligned} H(X) &= E\{I(x_j)\} \\ &= \sum_{j=1}^n P(x_j) \cdot I(x_j) \\ &= -\sum_{j=1}^n P(x_j) \cdot \log_{\alpha} P(x_j) \end{aligned} \quad (4-21)$$

$H(X)$  在信息论中称为信源  $X$  的“熵” (Entropy), 它的含义是信源  $X$  发出任意一个随机变量的平均信息量。

在 4-21 式中, 当  $\alpha$  取 2 时,  $H(X)$  的单位为比特 (bit);  $\alpha$  取  $e$  时,  $H(X)$  的单位为奈特 (Net);  $\alpha$  取 10 时,  $H(X)$  的单位为哈特 (Hart)。

图像编码中,  $\alpha$  取 2。

等概率事件的熵最大, 例如以  $N=8$  为例,  $P(x_1)=P(x_2)=P(x_3)=\dots=P(x_8)=\frac{1}{8}$  时,

$$H(X) = -\sum_{j=1}^8 \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits}$$

另外, 当  $P(x_1)=1$  时, 必然  $P(x_2)=P(x_3)=P(x_4)=P(x_5)=P(x_6)=P(x_7)=P(x_8)=0$ , 这时熵

$$H(X) = -P(x_1) \log_2 P(x_1) = 0$$

熵的范围为:

$$0 \leq H(X) \leq \log_2 N$$

在编码中用熵值衡量是否为最佳编码。若以  $\bar{L}$  表示编码器输出码字的平均码长, 则当

$\bar{L} \geq H(X)$  有冗余, 不是最佳;

$\bar{L} < H(X)$  不可能;

$\bar{L} = H(X)$  最佳编码 ( $\bar{L}$  稍大于  $H(X)$ )。

熵值是平均码长  $\bar{L}$  的下限。

#### 4.4.2 哈夫曼编码

哈夫曼编码方法于 1952 年问世, 迄今为止仍经久不衰, 广泛应用于各种数据压缩技术



中，且仍不失为熵编码中的最佳编码方法。

哈夫曼编码的理论依据是变字长编码理论。在变字长编码中，编码器的编码输出码字是字长不等的码字，按编码输入信息符号出现的统计概率，给输出码字分配以不同的字长。对于编码输入中，出现大概率的信息符号，赋以短字长的输出码字；对于编码输入中出现小概率的信息符号，赋以长字长的输出码字。可以证明，按照概率出现大小的顺序，对输出码字分配不同码字长度的变字长编码方法，其输出码字的平均码长最短，与信源熵值最接近，编码方法最佳。

哈夫曼编码的具体步骤归纳如下：

- 1) 概率统计（如对一幅图像或  $m$  幅同种类型图像作灰度信号统计）得到  $n$  个不同概率的信息符号。
- 2) 将  $n$  个信源信息符号的  $n$  个概率按概率大小排序。
- 3) 将  $n$  个概率中最后两个小概率相加，这时概率个数减为  $n-1$  个。
- 4) 将  $n-1$  个概率按大小重新排序。
- 5) 重复 3)，将新排序后的最后两个小概率再相加，相加和与其余概率再排序。
- 6) 如此反复重复  $n-2$  次，得到只剩两个概率序列。
- 7) 以二进制码元（0.1）赋值，构成哈夫曼码字。编码结束。

哈夫曼码字长度和信息符号出现概率大小次序正好相反，即大概信息符号分配码字长度短，小概率信息符号分配码字长度长。

下面以一个具体例子，说明哈夫曼编码过程，如图 4-11 和图 4-12。

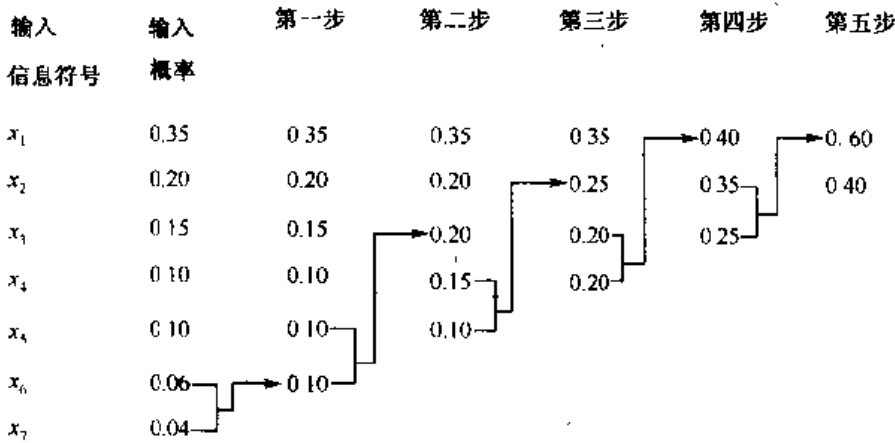


图 4-11 哈夫曼编码步骤

码字的平均码长  $\bar{N}$  以下面公式计算：

$$\begin{aligned}
 \bar{L} &= \sum_{j=1}^n P_j L_j \\
 &= \sum_{j=1}^7 (P_j L_j) \\
 &= (0.35 + 0.20) \times 2 + (0.15 + 0.10 + 0.10) \times 3 + (0.06 + 0.04) \times 4 \\
 &= 2.55 \text{bits / pel}
 \end{aligned}$$

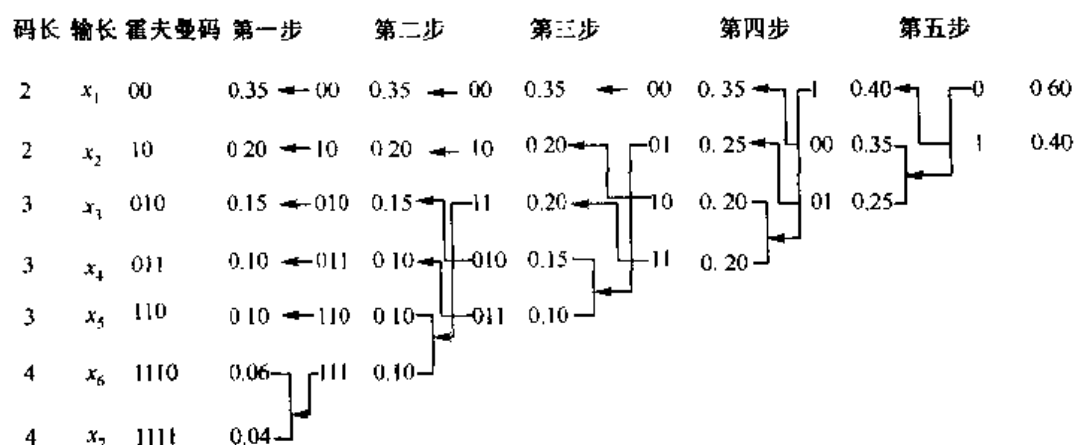


图 4-12 哈夫曼码字的构成

熵

$$\begin{aligned}
 H &= -\sum_{j=1}^n P(x_j) \cdot \log_2 P(x_j) \\
 &= -\sum_{j=1}^7 P(x_j) \cdot \log_2 P(x_j) \\
 &= -(0.35 \cdot \log_2 0.35 + 0.20 \cdot \log_2 0.20 + 0.15 \cdot \log_2 0.15 \\
 &\quad + (0.10 \cdot \log_2 0.10) \times 2 + 0.06 \cdot \log_2 0.06 + 0.04 \cdot \log_2 0.04) \\
 &= 2.13 \text{ bits/pel}
 \end{aligned}$$

通过这个例子，总结如下特点：

- 平均码长  $\bar{L} > H$  (熵)。
- 平均码长  $\bar{L} < 3 \text{ bits}$  (等长码需要的位数)。
- 保证解码的惟一性，短码字不构成长码字的前缀。
- 在接收端需保存一个与发送端相同的哈夫曼码表 (输入与哈夫曼码的对应表)。

### 4.4.3 算术编码

#### 1. 算术编码基本原理

算术编码方法比霍夫曼编码、行程长度等熵编码方法都复杂，但是它不需要传送像霍夫曼编码的霍夫曼码表，同时算术编码还有自适应能力的优点，所以算术编码是实现高效压缩数据中很有前途的编码方法。

算术编码初始化可预置两个参数， $P_e$  和  $Q_e$ 。 $P_e$  代表大概率， $Q_e$  代表小概率，以 MPS (Most Probable Symbol) 表示大概率符号，与  $P_e$  对应；以 LPS (Least Probable Symbol) 表示小概率符号，与  $Q_e$  对应。当符号流中“0”符号对应 MPS 和  $P_e$  时，则符号“1”对应 LPS 和  $Q_e$ ；反之当符号流中“1”符号对应 MPS 和  $P_e$  时，则符号“0”对应 LPS 和  $Q_e$ 。值得注意的是上述对应关系并不是一成不变的，随着被编码符号流中符号“0”和符号“1”出现的概率，上述关系将自适应地改变。

算术编码的第一步，根据概率  $Q_e$  和  $P_e$  值，将半开区间  $[0, 1)$  分割成两个子区间，如图 4-13 所示。 $Q_e$  为小概率，从 0 算起，那么  $P_e = 1 - Q_e$ 。

随后当编码输入符号流中，来的是“0”，其输出码字应落在  $0 \sim Q_c$  子区间内，在此区间内的一个最短二进制码作为输出；如果符号流中第一个符号来的是“1”，那么其输出码字应落在  $P_c$  段中，即  $Q_c \sim 1$  子区间内。此后当第二符号来临后，这时相当于对两个符号编码。方法是，对第一次分割结果的两个子区间，

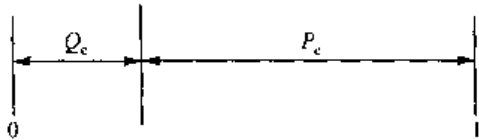


图 4-13  $[0, 1]$  区间分割

依  $Q_c$  和  $P_c$  概率进行再分割，得到 4 个子区间。第一个子区间是  $0 \sim Q_c Q_c$ ，第二个子区间是  $Q_c Q_c \sim Q_c$ ，第三个子区间是  $Q_c \sim Q_c + Q_c P_c$ ，第 4 个子区间是  $(Q_c + Q_c P_c) \sim 1$ 。当第二个符号来的是“0”，分割  $Q_c$  段，当第二个符号来的是“1”，对  $P_c$  段分割。第三个符号来临后，要区别以下情况，当第二个符号为“0”第三个符号也是“0”，则对  $0 \sim Q_c Q_c$  段依概率  $Q_c$ 、 $P_c$  分割，当第二个符号为“0”第三个符号为“1”，则对  $Q_c Q_c \sim Q_c$  段分割；当第二个符号为“1”第三个符号为“0”，则对  $Q_c \sim (Q_c + Q_c P_c)$  段分割，当第二个符号为“1”第三个符号为“1”，则对  $(Q_c + Q_c P_c) \sim 1$  最后一段分割。以此类推，直到一组符号结束为止。图 4-14 给出一个 4 个符号的分割区间图。由图看出，4 个输入符号可分割出 16 个子区间，子区间宽度不相等，所以落在不同子区间的输入可用不等长码字编码。表 4-1 给出与图 4-14 对应的 4 个符号的分割区间与码字分配表。表中列出每个区间起始点用二进制小数表示的数据，落入该区间的输入符号，以区间两端（起始，结束）二进制小数中的最小者表示。以小数点后的二进制码作为输出码字。但应注意，短码字不能成为长码字的前缀。

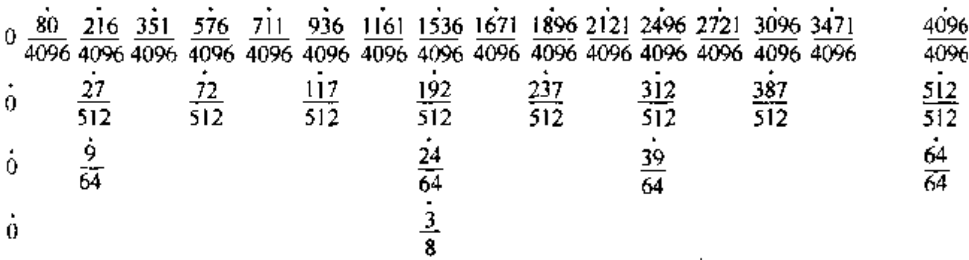


图 4-14 算术编码区间分割图

表 4-1 算术编码分割区间与码字

符 号	子 区 间	起 始 点	起 始 点 (用二进制小数表示)	落 入 子 区 间位数短的小数	码 字	子 区 间 宽
第 四 个 符 号	0000	0	0.00 00 00 00 00 00	0.0 00 00		81/4096
	0001	81/4096	0.00 00 01 01 00 01	0.0 00 01	00 00 1	135/4096
	0010	216/4096	0.00 00 11 01 10 00	0.0 00 1	00 01	135/4096
	0011	351/4096	0.00 01 01 01 11 11	0.0 01 00	00 10 0	225/4096
	0100	576/4096	0.00 10 01 00 00 00	0.0 01 01	00 10 1	135/4096
	0101	711/4096	0.00 10 11 00 01 11	0.0 01 1	00 11	225/4096
	0110	936/4096	0.00 11 10 10 10 00	0.0 10 0	01 00	225/4096
	0111	1161/4096	0.01 00 10 00 10 01	0.0 10 1	01 01	375/4096
	1000	1536/4096	0.01 10 00 00 00 00	0.0 11 0	01 10	135/4096
	1001	1671/4096	0.01 10 10 00 01 11	0.0 11 1	01 11	225/4096
	1010	1896/4096	0.01 11 01 10 10 00	0.1 00 0	10 00	225/4096

(续)

符 号	子 区 间	起 始 点	起始点 (用二进制小数表示)	落 入 子 区 间位数短的小数	码 字	子 区 间 宽
第 四 个 符 号	1011	2121/4096	0.100001001001	0.1001	1001	375/4096
	1100	2496/4096	0.100111000000	0.1010	1010	225/4096
	1101	2721/4096	0.101010100001	0.1011	1011	375/4096
	1110	3096/4096	0.110000011000	0.1101	1101	375/4096
	1111	3471/4096	0.110110001111	0.111	111	625/4096
		4096/4096	1.000000000000			

## 2. 编码算法及举例

以上为说明算术编码不断分割的原理，在图 4-14 和表 4-1 中将 4 个符号输入，所有可能的 16 个子区间的头、尾和其码字全部列出，实际不需要这样，实际问题是，只针对某个输入——“0”、“1”符号组合，求出其输出码字，也就是说求出这组符号将落入子区间的起点（头）和子区间的宽度，在这个范围内便可确定输出码字。下面介绍子区间头和宽的计算方法，并通过例子说明其过程。

编码时设置两个专用寄存器，A 寄存器和 C 寄存器，这两个寄存器中的内容是存储符号“0”或“1”到来之前子区间的状态参数。

设 C 寄存器内的数值为子区间的起始位置，A 寄存器内的数值为子区间的宽度，该宽度正好是已输入符号串的概率（初始化时 C=0，A=1）。

随着被编码符号流“0”符号和“1”符号不断地输入，C 寄存器中的值和 A 寄存器中的值，按以下规律不断修正。

当低概率符号 LPS 到来时

$$\begin{cases} C = C \\ A = AQ_e \end{cases} \quad (4-22)$$

当高概率符号 MPS 到来时

$$\begin{cases} C = C + A \cdot Q_e \\ A = AP_e = A(1 - Q_e) \end{cases} \quad (4-23)$$

C+A 等于子区间的右端点，算术编码的结果落在子区间内。输入编码符号串中大概率符号出现频率愈高，对应的子区间变宽，这时可用短的码字表示编码结果；相反，输入符号串中小概率符号出现频率增加，相应的子区间变窄，落入该区间的编码结果需要一个长的码字表示。下面我们举一个简单的例子，具体说明编码过程。

按以上规则，对一个“11011111”符号串进行算术编码。

分析得出“0”符号对应小概率符号，以 LPS 表示，其对应概率  $Q_e = (0.001)_b = (1/8)_d$ ；“1”为大概率符号，以 MPS 表示，其对应概率为  $P_e = (0.111)_b = (7/8)_d$ 。图 4-15 示出对符号串“11011111”算术编码过程。



符号	解码	判 $C'$ 落在	$C'$ 值计算	A 值计算	解码子区间
1	1	$QeA \rightarrow A$ 之间	$C' = C' - QeA$	$A = A(1 - Qe)$	$QeA = 0.001$ $C' = 0.0101$ $A = 1$
2	1	$QeA \rightarrow A$ 之间	$C' = C' - QeA$	$A = A(1 - Qe)$	$QeA = 0.000111$ $C' = 0.0011$ $A = 0.111$
3	0	$0 \rightarrow QeA$ 之间	$C' = C' - QeA$	$A = Qe$	$QeA = 0.000110001$ $C' = 0.000101$ $A = 0.110001$
4	1	$QeA \rightarrow A$ 之间	$C' = C' - QeA$	$A = A(1 - Qe)$	$QeA = 0.000000110001$ $A = 0.000110001$ $C' = 0.000101$
5	1	$QeA \rightarrow A$ 之间	$C' = C' - QeA$	$A = A(1 - Qe)$	$QeA$ $A = 0.000101010111$ $C' = 0.000100001111$
6	1	$QeA \rightarrow A$ 之间	$C' = C' - QeA$	$A = A(1 - Qe)$	$QeA$ $A = 0.000100001100001$ $C' = 0.000011100100001$
7	1	$QeA \rightarrow A$ 之间	$C' = C' - QeA$	$A = A(1 - Qe)$	$QeA$ $A = 0.000011101010100111$ $C' = 0.000010000100111$
8	1	$QeA \rightarrow A$ 之间	$C' = C' - QeA$	$A = A(1 - Qe)$	$QeA$ $A = 0.000011001101010010001$ $C' = 0.000010100101010010001$

图 4-16 算术解码原理图

当  $C'$  落在  $0 \sim QeA$  子区间内，解码符号赋以“0”，这时

$$\begin{cases} C' = C' \\ A = QeA \end{cases} \quad (4-24)$$

当  $C'$  落在  $QeA \sim A$  子区间内，解码符号赋以“1”，这时

$$\begin{cases} C' = C' - QeA \\ A = A(1 - Qe) \end{cases} \quad (4-25)$$

利用上述公式多次重复计算，求得与解码输入符号串所对应的解码输出。解码结果是由“0”、“1”构成的符号串。现在以上面编码结果“0101”为例，对它进行解码。

初始时，MPS 对应“1”符号，LPS 对应“0”符号， $Qe = (0.001)_2 = (1/8)d$ ， $A = 1$ ， $C' = 0.0101$ 。首先将区间  $[0, 1)$  分割成两个子区间，分割方法是让  $Qe$  靠近零的一侧， $Pe$  靠近 1 的一侧，以  $QeA$  为分界， $0 \sim Qe$  构成一个子区间， $QeA \sim A$  构成另一个子区间。解码开始后，判  $C' = 0.0101$  的值，落在哪个区间。

当  $C'$  落在  $0 \sim QeA$  子区间，解码符号赋以“0”，并计算：

$$\begin{cases} C' = C' \\ A = QeA \end{cases}$$

当  $C'$  落在  $QeA \sim A$  子区间解码符号赋以“1”，并计算：

$$\begin{cases} C' = C' - QeA \\ A = A(1 - Qe) \end{cases}$$

图 4-16 示出以 $(0.0101)_b$  为输入, 进行 8 次分割计算结果及其计算过程, 最后解码结果是“11011111”, 与图 4-15 编码输入完全相同。

4. 改进区间算法

图 4-15 和图 4-16 表明了算术编码和解码的工作原理。从这个例子的计算过程中看出, 编码过程中 A 值和 C 值及解码过程中的 C' 值和 A 值, 小数点后的位数愈来愈长, 这在实际中是无法实现的。下面介绍一种较实用的改进算法, 它的主要思想是限制小数点后的位数, 如图 4-17 和 4-18 所示分别表示编码和解码的计算过程。

(1) 改进区间编码

符号说明:

MIN: 输入符号比特流 (“0” 或 “1”)。

SK:  $Qe$  值的指数,  $Qe=2^{-SK}$ 。

E: 调整用指数。

$$W(F)=2^{-(E+SK)}=2^{-E} \times 2^{-SK}$$

(4-26)

$W(F)$ 是低概率  $Qe$  的函数。

$$W(T)=A-2^{-(E+SK)}=A-(2^{-E} \times 2^{-SK})$$

(4-27)

C: 起始点。

A: 宽度。

每一个编码符号到来后, 都要以上参数进行修改, A 值、C 值的计算列表见表 4-2。

表 4-2 改进区间编码参数修正表

输入符号	C	A
0	$C=C$	$A=W(F)$
1	$C=C+W(F)$	$A=W(T)=A-W(F)$

由图 4-17 改进区间算术编码图中看, 在编码过程中,

$$(0.11)_b \leq A \times 2^E < (1.1)_b \text{ 或 } (0.75)_d \leq A \times 2^E < (1.5)_d$$

其中指数 E 随 A 值的改变作调整, 使  $A \times 2^E$  的值落在半开区间 $[0.75, 1.5)$ 之间。

(2) 改进区间解码

改进区间解码如图 4-18 所示, 在整个过程中, 保证 $(0.11)_b \leq A' < (1.1)_b$  或 $(0.75)_d \leq A' < (1.5)_d$

判决条件及 A' 值, C' 值计算列表如表 4-3。

表 4-3 改进区间解码参数修正表

判决条件	解码	E'	C' 值	A' 值
$C' < 2^{-SK}$	0	SK	$C' = 2^{SK} C'$	$A' = 1$
$C' \geq 2^{-SK}$	1	随 A' 值改变保证 A' 落入 $[0.75, 1.5)$ 之间	$C' = 2^{E'} (C' - 2^{-SK})$	$A' = 2^{E'} (A' - 2^{-SK})$

事件	输入 MIN	SK	W(F)	E	C	A	子区间图: 保证 $(0.11)_b \leq A \cdot 2^K < (1.1)$
初始 状态	—	—	0	—	0.00000000	1.00000000	
1	1	3	0	0.001	0.01000000	0.11100000	
2	1	3	0	0.001	0.01000000	0.11000000	
3	0	3	0	0.001	0.01000000	0.00100000	
4	1	3	3	0.000001	0.01000100	0.00011100	
5	1	3	3	0.000001	0.01001100	0.00011000	
6	1		3	0.000001	0.01001110	0.00010100	
7	1	3	4	0.0000001	0.010110000	0.00010010	
8	1	3	4	0.0000001	0.01010000 头	0.00010000	

图 4-17 改进区间算术编码

序号	解 码	判决条件	E'	C' 值计算 $C' = \begin{cases} 2^{SK} C' & C' < 0.001 \\ 2^K (C' - 2^{-SK}) & C' \geq 0.001 \end{cases}$	A' 值计算 $A' = \begin{cases} 1 & C' < 0.001 \\ 2^K (A - 2^{-SK}) & C' \geq 0.001 \end{cases}$	扩大的子区间图: 保证 $(0.11)_b \leq A' < (1.1)_b$ $(0.75)_d \quad (1.5)_d$
1	1	$C' \geq 0.001$	0	0.0101	1	
2	1	$C' \geq 0.001$	0	0.0011	0.111	
3	0	$C' \geq 0.001$	0	0.0001	0.110	
4	1	$C' \geq 0.001$	3	0.1000	1	
5	1	$C' \geq 0.001$	0	0.0110	0.111	
6	1	$C' \geq 0.001$	0	0.0100	0.110	
7	1	$C' \geq 0.001$	1	0.0100	1.010	
8	1	$C' \geq 0.001$	0	0.0010	1.001	

图 4-18 改进区间算术解码

算术编码和哈夫曼编码都是根据统计来确定估计概率的，当实际编码信号的概率与估计概率一致时会得到最好的压缩，并且理论上哈夫曼编码优于算术编码。但是实际上估计概率与信号的实际概率很难一致，并且算术编码的估计概率随着信号概率变化而改变，因此算术编码的编码效率有时优于哈夫曼编码。

JPEG 建议了两种熵编码方法——哈夫曼编码和算术编码。JPEG 成员测试过，对于许



多图像, 算术编码的压缩效果比哈夫曼的压缩效果要好 5%~10%。

## 4.5 多媒体数据压缩编码的国际标准

从 20 世纪 80 年代开始, 世界上已有几十家公司纷纷投入到多媒体计算机系统的研制和开发工作。20 世纪 90 年代初已有不少精彩的多媒体产品问世。具有人机交互特色的多媒体技术, 使久居科学殿堂、供专业人员使用的计算机, 进入普通人的家庭, 进入人们的生活、学习、娱乐及人们的精神生活领域。人们像使用家用电器一样地使用计算机。计算机能听懂人的讲话、计算机也能讲话的实用型产品, 在不久的将来, 也会进入市场。

Internet 技术的迅猛发展与普及, 推动了世界范围的信息传输和信息交流, 波澜壮阔的信息化浪潮, 将席卷全球。

在色彩缤纷、变幻无穷的多媒体世界中, 用户如何选择产品, 用户能自由地, 组合、装配来自不同厂家的产品部件, 构成自己满意的系统? 这就提出了一个不同厂家产品的兼容性问题, 因此需要一个全球性的统一国际技术标准。

国际标准化协会 (International Standardization Organization——ISO), 国际电子学委员会 (International Electronics Committee——IEC), 国际电信协会 (International Telecommunication Union——ITU) 等国际组织, 于 20 世纪 90 年代制定了三个重要的多媒体国际标准。这三个标准是 JPEG 标准、MPEG 标准和 H.261 标准。

### 4.5.1 JPEG 标准

#### 1. 引言

JPEG (Joint Photographic Experts Group) 是联合图像专家小组的英文缩写。其中“联合”的含意是指, 国际电报电话咨询委员会 (CCITT) 和国际标准化协会 (ISO) 联合组成的一个图像专家小组。联合图像专家小组多年来一直致力于标准化工作, 他们开发研制出连续色调、多级灰度、静止图像的数字图像压缩编码方法。这个压缩编码方法称为 JPEG 算法。JPEG 算法被确定为 JPEG 国际标准, 它是国际上彩色、灰度、静止图像的第一个国际标准。JPEG 标准是一个适用范围广泛的通用标准。它不仅适于静图像的压缩, 电视图像序列的帧内图像的压缩编码也常采用 JPEG 压缩标准。

JPEG 的目的是为了给出一个适用于连续色调图像的压缩方法, 使之满足以下要求:

(1) 达到或接近当前压缩比与图像保真度的技术水平, 能覆盖一个较宽的图像质量等级范围, 能达到“很好”到“极好”的评估, 与原始图像相比, 人的视觉难以区分。

(2) 能适用于任何种类的连续色调的图像, 且长宽比都不受限制, 同时也不受限于景物内容、图像的复杂程度和统计特性等。

(3) 计算的复杂性是可控制的, 其软件可在各种 CPU 上完成, 算法也可用硬件实现。

(4) JPEG 算法, 具有以下四种操作方式:

- 顺序编码。每一个图像分量按从左到右, 从上到下扫描, 一次扫描完成编码。
- 累进编码。图像编码在多次扫描中完成。累进编码传输时间长, 接收端收到的图像是多次扫描由粗糙到清晰的累进过程。
- 无失真编码。无失真编码方法保证解码后完全精确地恢复源图像采样值, 其压缩比

低于有失真压缩编码方法。

- 分层编码。图像在多个空间分辨率进行编码。在信道传送速率慢，接收端显示器分辨率也不高的情况下，只需做低分辨率图像解码。

## 2. 基于离散余弦变换的编码

下面举一个基于离散余弦变换（DCT）的有失真 JPEG 编解码的示例，了解掌握 JPEG 编解码的全过程。图 4-19 是基于 DCT JPEG 编码的过程框图，图 4-20 是解码过程框图。这两个图表示的是一个单分量（如图像的灰度信号）图像的压缩编码和解码过程，表示出基于 DCT 顺序工作方式编解码的完整工作过程。对于彩色图像，可以近似看作多分量（亮度信号分量和色度信号分量等）进行压缩和解压缩处理。

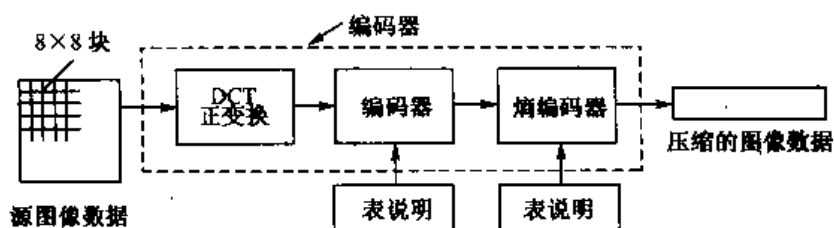


图 4-19 基于 DCT 编码器处理步骤

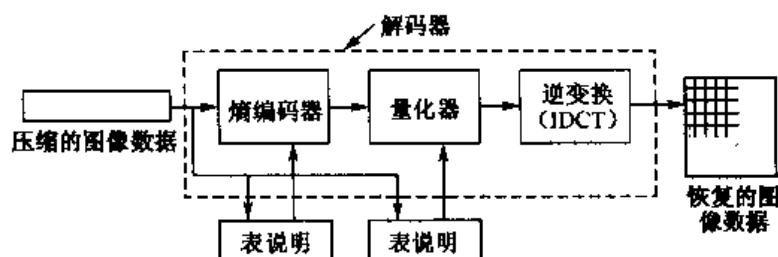


图 4-20 基于 DCT 解码器处理步骤

### (1) 离散余弦变化 (DCT)

JPEG 采用的是  $8 \times 8$  大小的子块的二维离散余弦变换 DCT (Discrete Cosine Transform)。

在编码器的输入端，把原始图像顺序地分割成一系列  $8 \times 8$  的子块，设原始图像的采样精度为  $P$  位，是无符号整数，输入时把  $[0, 2^P-1]$  范围的无符号整数变成  $[-2^{P-1}, 2^{P-1}-1]$  范围的有符号整数，以此作为离散余弦正变换 FDCT (Forward DCT) 的输入。在解码器的输出端经离散余弦逆变换 (IDCT) (Inverse DCT) 后，得到一系列  $8 \times 8$  的图像数据块，需将其数值范围由  $[-2^{P-1}, 2^{P-1}-1]$  再变回到  $[0, 2^P]$  范围的无符号整数，来获得重构图像。

下面的公式是  $8 \times 8$  FDCT 和  $8 \times 8$  IDCT 数学定义表达式。

正变换：

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[ \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cdot \cos \frac{(2x+1)u\pi}{16} \cdot \cos \frac{(2y+1)v\pi}{16} \right] \quad (4-28)$$

逆变换：

$$f(x, y) = \frac{1}{4} \left[ \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cdot \cos \frac{(2x+1)}{16} u\pi \cdot \cos \frac{(2y+1)}{16} v\pi \right] \quad (4-29)$$

其中:

$$\begin{cases} C(u), C(v) = 1/\sqrt{2} & \text{当 } u, v=0 \text{ 时} \\ C(u), C(v) = 1 & \text{其他} \end{cases}$$

从二维 DCT 的计算公式看出, 它们具有可分离的变换特征, 所以二维 DCT 可分解成行向的一维 DCT 计算和列向的一维 DCT 计算的组合运算。二维快速余弦变换 (2-FDCT) 是把  $8 \times 8$  块不断分成更小的无交叠子块, 直接对数据块进行运算操作。

对基于 DCT 压缩算法的简单而直观的认识, 可把 FDCT 看作一个谐波分析仪和把 IDCT 看作一个谐波合成器。 $8 \times 8$  数据块输入分解成 64 个正交基信号, 每个正交基信号对应于 64 个独立二维空间频率中的一个, 这些空间频率是由输入信号的“频谱”组成。FDCT 输出 64 个基信号的幅值称作“DCT 系数”, 即 DCT 变换系数值。64 个变换系数中包括一个代表直流分量的“DC 系数”和 63 个代表交流分量的“AC 系数”。IDCT 是 FDCT 的逆过程, 它把 64 个 DCT 变换系数经逆变换运算, 重建一个 64 点的输出图像。如果 FDCT 和 IDCT 变换计算所使用的设备的计算精度足够高, 且系数未经过量化, 那么原始的 64 点信号就能精确地恢复。

## (2) 量化

为了达到压缩数据的目的, 对 DCT 系数  $F(u, v)$  需作量化处理。量化处理是一个多到一的映射, 它是造成 DCT 编解码信息损失的根源。在 JPEG 标准中采用线性均匀量化器, 量化定义为对 64 个 DCT 系数除以量化步长, 四舍五入取整, 如 4-30 公式所示。

$$F^Q(u, v) = \text{Integer Round}(F(u, v)/Q(u, v)) \quad (4-30)$$

其中  $Q(u, v)$  是量化器步长。它是量化表的元素, 量化表元素随 DCT 变换系数的位置和彩色分量的不同有不同值。量化表的尺寸为  $8 \times 8$ , 与 64 个变换系数一一对应。这个量化表应该由用户规定 (在 JPEG 标准中给出参考值), 并作为编码器的一个输入。量化表中的每个元素值为 1 到 255 之间的任意整数, 其值规定了它所对应 DCT 系数的量化器步长。

反量化的计算公式如 4-31 式所示。

$$F^Q(u, v) = F(u, v) \times Q(u, v) \quad (4-31)$$

量化的作用是在一定的主观保真度图像质量的前提下, 丢掉那些对视觉效果影响不大的信息。不同频率的余弦函数对视觉影响不同, 所以可根据不同频率的视觉阈值来选择量化表中的元素值的大小。这样通过心理视觉实验可确定对应于不同频率的视觉阈值, 以确定不同频率的量化器步长。图 4-21 示出量化特性图。

表 4-4 给出一个根据心理视觉加权函数而得的亮度分量量化矩阵, 即亮度量化表。表 4-5 给出根据心理视觉加权函数而得到的色度分量量化矩阵, 即色度量化表, 是 JPEG 中给出的参考值。

DCT 变换系数  $F(u, v)$  除以量化表中对应位置的量化步长, 其幅值下降, 高频系数的零值数目增加。

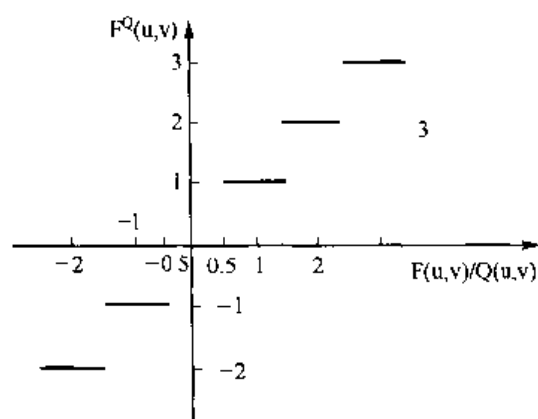


图 4-21 量化特性

表 4-4 亮度量化表

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

表 4-5 色度量化表

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

### (3) DC 系数编码和 AC 系数的行程编码

64 个变换系数经量化后，坐标  $u = v = 0$  的 DC 系数是直流分量，即为 64 个空域图像采样值的平均值。相邻  $8 \times 8$  块之间的 DC 系数有强的相关性，JPEG 中对 DC 系数采用 DPCM 编码，或差分编码，即对相邻块之间的 DC 系数差值  $DIFF = DC_j - DC_{j-1}$  编码。如图 4-22 所示。

其余 63 个交流系数采用行程编码。从左上方  $AC_{01}$  开始，沿对角线方向，以“Z”字形行程扫描，直到  $AC_{77}$  扫描结束。如图 4-23 所示。量化后待编码的 AC 系数通常有许多零值，沿“Z”字形路径进行

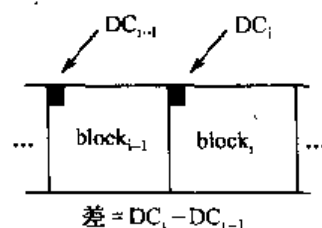


图 4.22 DC 系数差分编码

行程编码，可增加行程中连续零的个数。63 个 AC 系数行程编码的码字，可用两个字节表示。如图 4-24 所示。

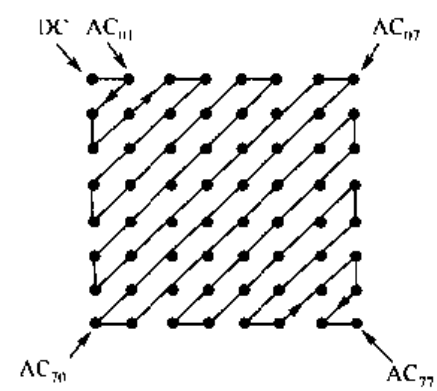


图 4-23 “Z”字形排列

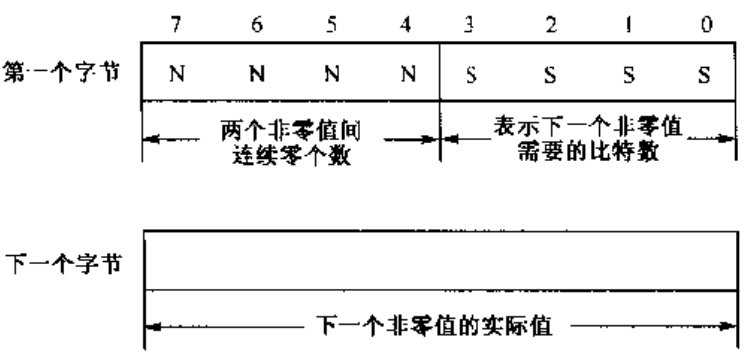


图 4-24 AC 系数行程编码码字

(4) 熵编码

为了进一步达到压缩数据的目的，需要对量化后的 DC 码和 AC 行程编码的码字再作基于统计特性的熵编码。JPEG 建议使用两种熵编码方法：哈夫曼（Huffman）编码和自适应二进制算术编码（Adaptive Binary Arithmetic Coding）。熵编码可分成两步进行，首先把 DC 码和行程码字转换成一个中间符号序列，然后给这些符号赋以变长码字。

1) 熵编码中间格式表示。熵编码的中间格式由两个符号组成。

符号 1（行程，尺寸）

符号 2（幅值）

符号 1 中的第一个信息参数“行程”，表示前后两个非零 AC 系数之间连续零的个数；第二个信息参数“尺寸”是后一个非零 AC 系数幅值表示需要的位数。一个基本符号 1，可表示的行程范围为 1 到 15，其结构于表 4-6 示出，当两个非零 AC 系数之间连续零的个数超过 15 时，用增加扩展符号 1 “(15, 0)” 的个数来扩充。对于 8×8 块的 63 个 AC 系数最多增加三个 “(15, 0)” 扩展符号 1。符号 2 中的幅值代表非零 AC 系数的数值，其范围为[-2<sup>10</sup>, 2<sup>10</sup>-1]，结构形式如表 4-7 所示。

表 4-6 基本哈夫曼编码符号-1 结构

	大 小					
	0	1	2	...	9	10
0	EOB					
•	×					
RUN LENGTH	×				RUN SIZE	
•	×				VALUES	
15	ZRL					

表 4-7 基本熵编码符号-2 结构

大 小	范 围
1	-1,1
2	-3,-2,2,3
3	-7,-4,4,7
4	-15,-8,8,15
5	-31,-16,16,31
6	-63,-32,32,63
7	-127,-64,64,127
8	-255,-128,128,255
9	-511,-256,256,511
10	-1023,-512,512,1023

对于  $8 \times 8$  块的直流分量 DC 系数，也以类似于 AC 系数的符号 1 和符号 2 格式表示。只是 DC 系数的符号 1 只代表尺寸信息，用以表示 DC 差值的幅值所需的位数；符号 2 表示差值的幅值，其动态范围为  $[-2^{11}, 2^{11}-1]$ ，在表 4-7 的末尾再多加一级，所以幅值尺寸以 1 到 11 位表示。

2) 可变长度熵编码。熵编码的下一步工作是将 63 个 AC 系数表示成为符号 1 和符号 2 对的序列。零行程长度超过 15，有多个符号 1，块结束 (EOB) 时只有一个符号 1 以 (0, 0) 表示。可变长度熵编码就是针对这种符号序列进行编码。对 DC 系数和 AC 系数中的符号 1 采用哈夫曼表中的变长码 (variable-length code 即 VLC) 编码。哈夫曼变长码表必须作为 JPEG 编码器的输入。需要注意的是在数据流中哈夫曼表的表示格式是一个间接的说明，在解码时，解码器利用这一间接说明重构真正的哈夫曼表。符号 2 用码字长度已在表 4-7 给出的变长整数 (Variable-Length Integer, 即 VLI) 码编码。VLI 是变长码但不是哈夫曼码，VLI 的长度存放在 VLC 中，VLI 的码字被固化在 JPEG 建议中，供计算用。JPEG 解码器最多能够同时存放 4 套不同的熵编码表。表 4-8 和表 4-9 分别给出直流系数 DC 的亮度分量典型的哈夫曼表和 DC 的色度分量典型的哈夫曼表。

表 4-8 亮度 DC 系数表

分 类	码 长	码 字
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

表 4-9 色度 DC 系数表

分 类	码 长	码 字
0	2	00
1	2	01
2	2	10
3	3	110
4	4	1110
5	5	11110
6	6	111110
7	7	1111110
8	8	11111110
9	9	111111110
10	10	1111111110
11	11	11111111110

典型的直流分量（DC）差分编码表的说明如下。

下面 16 个字节是说明亮度 DCT 系数表的码字长度表：

x' 00 01 05 01 01 01 01 01 01 00 00 00 00 00 00 00'

紧跟在 16 个字节之后是一组值（说明亮度表的分类）：

x' 00 01 02 03 04 05 06 07 08 09 0A 0B'

下面 16 个字节是说明色度 DC 系数表的码字长度表：

x' 00 03 01 01 01 01 01 01 01 01 01 00 00 00 00 00'

紧跟在 16 个字节之后的是一组数值（说明色度表的分类）：

x' 00 01 02 03 04 05 06 07 08 09 0A 0B'

表 4-10 和表 4-11 分别给出交流系数 AC 的亮度分量典型的哈夫曼表和 AC 系数色度分量典型的哈夫曼表。

表 4-10 亮度 AC 系数表

行程/尺寸	码 长	码 字	行程/尺寸	码 长	码 字
0/0(EOB)	4	1010	1/1	4	1100
0/1	2	00	1/2	5	11011
0/2	2	01	1/3	7	1111001
0/3	3	100	1/4	9	111110110
0/4	4	1011	1/5	11	11111110110
0/5	5	11010	1/6	16	1111111110000100
0/6	7	1111000	1/7	16	1111111110000101
0/7	8	11111000	1/8	16	1111111110000110
0/8	10	1111110110	1/9	16	1111111110000111
0/9	16	1111111110000010	1/A	16	1111111110001000
0/A	16	1111111110000011	2/1	5	11100

(续)

行程/尺寸	码 长	码 字	行程/尺寸	码 长	码 字
2/2	8	11111001	6/2	12	111111101110
2/3	10	1111110111	6/3	16	111111110100110
2/4	12	11111110100	6/4	16	111111110100111
2/5	16	111111110001001	6/5	16	111111110101000
2/6	16	111111110001010	6/6	16	111111110101001
2/7	16	111111110001011	6/7	16	111111110101010
2/8	16	111111110001100	6/8	16	111111110101011
2/9	16	111111110001101	6/9	16	111111110101100
2/A	16	111111110001110	6/A	16	111111110101101
3/1	6	111010	7/1	8	11111010
3/2	9	111110111	7/2	12	11111110111
3/3	12	11111110101	7/3	16	111111110101110
3/4	16	111111110001111	7/4	16	111111110101111
3/5	16	111111110010000	7/5	16	111111110110000
3/6	16	111111110010001	7/6	16	111111110110001
3/7	16	111111110010010	7/7	16	111111110110010
3/8	16	111111110010011	7/8	16	111111110110011
3/9	16	111111110010100	7/9	16	111111110110100
3/A	16	111111110010101	7/A	16	111111110110101
4/1	6	111011	8/1	9	111111000
4/2	10	1111111000	8/2	15	11111111000000
4/3	16	111111110010110	8/3	16	111111110110110
4/4	16	111111110010111	8/4	16	111111110110111
4/5	16	111111110011000	8/5	16	111111110111000
4/6	16	111111110011001	8/6	16	111111110111001
4/7	16	111111110011010	8/7	16	111111110111010
4/8	16	111111110011011	8/8	16	111111110111011
4/9	16	111111110011100	8/9	16	111111110111100
4/A	16	111111110011101	8/A	16	111111110111101
5/1	7	1111010	9/1	9	111111001
5/2	11	11111110111	9/2	16	111111110111110
5/3	16	111111110011110	9/3	16	111111110111111
5/4	16	111111110011111	9/4	16	111111111000000
5/5	16	111111110100000	9/5	16	111111111000001
5/6	16	111111110100001	9/6	16	111111111000010
5/7	16	111111110100010	9/7	16	111111111000011
5/8	16	111111110100011	9/8	16	111111111000100
5/9	16	111111110100100	9/9	16	111111111000101
5/A	16	111111110100101	9/A	16	111111111000110
6/1	7	1111011	A/1	9	111111010



(续)

行程/尺寸	码 长	码 字	行程/尺寸	码 长	码 字
A/2	16	111111111000111	D/2	16	111111111100010
A/3	16	1111111111001000	D/3	16	1111111111100011
A/4	16	1111111111001001	D/4	16	1111111111100100
A/5	16	1111111111001010	D/5	16	1111111111100101
A/6	16	1111111111001011	D/6	16	1111111111100110
A/7	16	1111111111001100	D/7	16	1111111111100111
A/8	16	1111111111001101	D/8	16	1111111111101000
A/9	16	1111111111001110	D/9	16	1111111111101001
A/A	16	1111111111001111	D/A	16	1111111111101010
B/1	10	1111111001	E/1	16	111111111101011
B/2	16	111111111010000	E/2	16	1111111111101100
B/3	16	1111111111010001	E/3	16	1111111111101101
B/4	16	1111111111010010	E/4	16	1111111111101110
B/5	16	1111111111010011	E/5	16	1111111111101111
B/6	16	1111111111010100	E/6	16	111111111110000
B/7	16	1111111111010101	E/7	16	111111111110001
B/8	16	1111111111010110	E/8	16	111111111110010
B/9	16	1111111111010111	E/9	16	111111111110011
B/A	16	1111111111011000	E/A	16	111111111110100
C/1	10	1111111010	F/0(ZRL)	11	1111111001
C/2	16	111111111011001	F/1	16	111111111110101
C/3	16	1111111111011010	F/2	16	111111111110100
C/4	16	1111111111011011	F/3	16	111111111110111
C/5	16	1111111111011100	F/4	16	11111111111000
C/6	16	1111111111011101	F/5	16	11111111111001
C/7	16	1111111111011110	F/6	16	11111111111010
C/8	16	1111111111011111	F/7	16	11111111111011
C/9	16	111111111100000	F/8	16	111111111111100
C/A	16	111111111100001	F/9	16	111111111111101
D/1	11	1111111000	F/A	16	111111111111110

表 4-11 色度 AC 系数表

行程/尺寸	码 长	码 字	行程/尺寸	码 长	码 字
0/0(EOB)	2	00	0/7	7	1111000
0/1	2	01	0/8	9	111110100
0/2	3	100	0/9	10	1111110110
0/3	4	1010	0/A	12	11111110100
0/4	5	11000	1/1	14	1011
0/5	5	11001	1/2	6	111001
0/6	6	111000	1/3	8	11110110

(续)

行程/尺寸	码 长	码 字	行程/尺寸	码 长	码 字
1/4	9	111110101	5/5	16	111111110100001
1/5	11	1111110110	5/6	16	111111110100010
1/6	12	11111110101	5/7	16	111111110100011
1/7	16	111111110001000	5/8	16	111111110100100
1/8	16	111111110001001	5/9	16	111111110100101
1/9	16	111111110001010	5/A	16	111111110100110
1/A	16	111111110001011	6/1	7	1111001
2/1	5	11010	6/2	11	1111110111
2/2	8	11110111	6/3	16	111111110100111
2/3	10	111110111	6/4	16	111111110101000
2/4	12	11111110110	6/5	16	111111110101001
2/5	15	11111111000010	6/6	16	111111110101010
2/6	16	111111110001100	6/7	16	111111110101011
2/7	16	111111110001101	6/8	16	111111110101100
2/8	16	111111110001110	6/9	16	111111110101101
2/9	16	111111110001111	6/A	16	111111110101110
2/A	16	111111110010000	7/1	7	1111010
3/1	5	11011	7/2	11	1111111000
3/2	8	11111000	7/3	16	111111110101111
3/3	10	111111000	7/4	16	111111110110000
3/4	12	11111110111	7/5	16	111111110110001
3/5	16	111111110010001	7/6	16	111111110110010
3/6	16	111111110010010	7/7	16	111111110110011
3/7	16	111111110010011	7/8	16	111111110110100
3/8	16	111111110010100	7/9	16	111111110110101
3/9	16	111111110010101	7/A	16	111111110110110
3/A	16	111111110010110	8/1	8	11111001
4/1	6	111010	8/2	16	111111110110111
4/2	9	11110110	8/3	16	111111110111000
4/3	16	111111110010111	8/4	16	111111110111001
4/4	16	111111110011000	8/5	16	111111110111010
4/5	16	111111110011001	8/6	16	111111110111011
4/6	16	111111110011010	8/7	16	111111110111100
4/7	16	111111110011011	8/8	16	111111110111101
4/8	16	111111110011100	8/9	16	111111110111110
4/9	16	111111110011101	8/A	16	111111110111111
4/A	16	111111110011110	9/1	9	111110111
5/1	6	111011	9/2	16	111111111000000
5/2	10	1111111001	9/3	16	111111111000001
5/3	16	111111110011111	9/4	16	111111111000010
5/4	16	111111110100000	9/5	16	111111111000011

(续)

行程/尺寸	码 长	码 字	行程/尺寸	码 长	码 字
9/6	16	111111111000100	C/9	16	111111111100010
9/7	16	111111111000101	C/A	16	111111111100011
9/8	16	111111111000110	D/1	11	1111111001
9/9	16	111111111000111	D/2	16	111111111100100
9/A	16	111111111001000	D/3	16	111111111100101
A/1	9	111111000	D/4	16	111111111100110
A/2	16	111111111001001	D/5	16	111111111100111
A/3	16	111111111001010	D/6	16	111111111101000
A/4	16	111111111001011	D/7	16	111111111101001
A/5	16	111111111001100	D/8	16	111111111101010
A/6	16	111111111001101	D/9	16	111111111101011
A/7	16	111111111001110	D/A	16	111111111101100
A/8	16	111111111001111	E/1	14	1111111100000
A/9	16	111111111010000	E/2	16	111111111101101
A/A	16	111111111010001	E/3	16	111111111101110
B/1	9	111111001	E/4	16	111111111101111
B/2	16	111111111010010	E/5	16	111111111110000
B/3	16	111111111010011	E/6	16	111111111110001
B/4	16	111111111010100	E/7	16	111111111110010
B/5	16	111111111010101	E/8	16	111111111110011
B/6	16	111111111010110	E/9	16	111111111110100
B/7	16	111111111010111	E/A	16	111111111110101
B/8	16	111111111011000	F/0(ZRL)	10	1111111010
B/9	16	111111111011001	F/1	15	11111111000011
B/A	16	111111111011010	F/2	16	111111111110110
C/1	9	111111010	F/3	16	111111111110111
C/2	16	111111111011011	F/4	16	111111111111000
C/3	16	111111111011100	F/5	16	111111111111001
C/4	16	111111111011101	F/6	16	111111111111010
C/5	16	111111111011110	F/7	16	111111111111011
C/6	16	111111111011111	F/8	16	111111111111100
C/7	16	111111111100000	F/9	16	111111111111101
C/8	16	111111111100001	F/A	16	111111111111110

交流分量 (AC 系数) 的哈夫曼表的说明如下。

下面 16 个字节是说明亮度 AC 系数表的码字长度表:

X' 00 02 01 03 03 02 04 03 05 05 04 04 00 00 01 7D'

16 个字节之后紧跟着一组数值 (此值对应于亮度 AC 系数符号 1 的值):

X' 01 02 03 00 04 11 05 12 21 31 41 06 13 51 61 07

22	71	14	32	81	91	A1	08	23	42	B1	C1	15	52	D1	F0
24	33	62	72	82	09	0A	16	17	18	19	1A	25	26	27	28
29	2A	34	35	36	37	38	39	3A	43	44	45	46	47	48	49
4A	53	54	55	56	57	58	59	5A	63	64	65	66	67	68	69
6A	73	74	75	76	77	78	79	7A	83	84	85	86	87	88	89
8A	92	93	94	95	96	97	98	99	9A	A2	A3	A4	A5	A6	A7
A8	A9	AA	B2	B3	B4	B5	B6	B7	B8	B9	BA	C2	C3	C4	C5
C6	C7	C8	C9	CA	D2	D3	D4	D5	D6	D7	D8	D9	DA	E1	E2
E3	E4	E5	E6	E7	E8	E9	EA	F1	F2	F3	F4	F5	F6	F7	F8
F9	FA														

下面 16 个字节是说明色度系数表的码字长度表：

X'00 02 01 02 04 04 03 04 07 05 04 04 00 01 02 77'

16 个字节之后紧跟着一组数值（此值对应于前面色度 AC 系数符号 1 的值）

X'00	01	02	03	11	04	05	21	31	06	12	41	51	07	61	71
13	22	32	81	08	14	42	91	A1	B1	C1	09	23	33	52	F0
15	62	72	D1	0A	16	24	34	E1	25	F1	17	18	19	1A	26
27	28	29	2A	35	36	37	38	39	3A	43	44	45	46	47	48
49	4A	53	54	55	56	57	58	59	5A	63	64	65	66	67	68
69	6A	73	74	75	76	77	78	79	7A	82	83	84	85	86	87
88	89	8A	92	93	94	95	96	97	98	99	9A	A2	A3	A4	A5
A6	A7	A8	A9	AA	B2	B3	B4	B5	B6	B7	B8	B9	BA	C2	C3
C4	C5	C6	C7	C8	C9	CA	D2	D3	D4	D5	D6	D7	D8	D9	DA
E2	E3	E4	E5	E6	E7	E8	E9	EA	F2	F3	F4	F5	F6	F7	F8
F9	FA														

### （5）压缩比和图像质量

基于 DCT 的 JPEG 压缩算法，对于中等复杂程度的彩色图像的压缩比与恢复图像的质量列表如表 4-12。

表 4-12 压缩比与恢复图像的质量

压缩效果（位/像素）	质 量
0.25~0.50	中~好，满足某些应用
0.50~0.75	好~很好，满足多数应用
0.75~1.5	极好，满足大多数应用
1.5~2.0	与原始图像分不出

为了更清楚地了解 JPEG 编码过程，给一个亮度子块编码的示例：设亮度子块按 Z 序排列的系数如下：

K	0	1	2	3	4	5	6	7	8	9~30	31	32~63
系数	12	5	-2	0	2	0	0	0	1	0	-1	0

请按 JPEG 基本系统编码给出该子块的编码。

1) 对于 DC 系数 12，因落入（-15~-8，8~15）范围，查表 4-7 得到 4；再查表 4-8，得

到码字为 101; 因为  $12 > 0$ , 故 4 位附加位可直接写出 12 的二进制码: “1100”, 因此  $ZZ(0)=12$  的编码为 “1011100”。

2)  $ZZ(1)=5$ , 它与  $ZZ(0)$  之间无零系数, 故  $NNNN=0$ , 因 5 落入表 4-8 的第 3 组, 所以  $SSSS=3$ , 而  $NNNN/SSSS=0/3$  的哈夫曼码可由表 4-10 亮度 AC 系数查得 “100”, 故  $ZZ(1)=5$  的编码为 “100101”。

3)  $ZZ(2)=-2$ ,  $NNNN/SSSS=0/2$ , 查表 4-10, 为 “01”, 而  $ZZ(2)-1=-3$ , 其码字低位二进制数为 “01”, 所以  $ZZ(2)=-2$  的编码为 “0101”。

$ZZ(K)$  中非零值的实际值的编码规定如下: 若  $ZZ(K) > 0$ , 附加位为  $ZZ(K)$  的最低 B 位, 如 (1), (2) 中所示; 若  $ZZ(K) < 0$ , 则为  $ZZ(K)-1$  的最低 B 位, 如 (3) 中所示。

4)  $ZZ(3)=0$ ,  $ZZ(4)=2$  中间有一个零系数, 即  $NNNN/SSSS=1/2$ , 查表 4-10 得到 “11011”, 而 2 的二进制码字为 “10”, 所以  $ZZ(3) \sim ZZ(4)$  的编码为 “1101110”。

5)  $ZZ(5) \sim ZZ(7)=0$ ,  $ZZ(8)=1$ , 则  $NNNN/SSSS=3/1$ , 查表 4-10 得到 “111010”, 故  $ZZ(5) \sim ZZ(8)$  的编码为 “1110101”。

6)  $ZZ(9) \sim ZZ(30)$ ,  $ZZ(31)=-1$ , 由于  $NNNN=30-9+1=22 > 15$ , 故先编一个 F/O 码, 从表 4-10 可查得其码字为 “1111111001”; 此后有  $NNNN=22-16=6 < 15$ , 故再编码  $NNNN/SSSS=6/1$ , 查表 4-10 得到码字为 “1111011”, 而  $-1-1=-2$  的编码为 “0”, 所以  $ZZ(9) \sim ZZ(31)$  的编码为: “1111111001+1111011+0”。

7)  $ZZ(32) \sim ZZ(63)=0$ , 由于  $ZZ(63)=0$ , 所以直接用 “EOB(%)” 结束本子块, 其码字查表 4-10 为 “1010”; 综合 (1) ~ (7), 可知本子块的编码比特流为:

“1011100+100101+0101+1101110+1110101+1111111001+11110110+1010”

共用了 54bits, 而原始图像子块要用  $8 \times 8 \times 8=512$ bits 表示, 压缩比为  $512:54 \approx 9.48:1$ 。

## 4.5.2 MPEG-1 标准

### 1. 引言

MPEG (Moving Picture Experts Group) 运动图像专家小组的活动开始于 1988 年, 其目标是要在 1990 年建立一个标准的草案。MPEG 和 JPEG 两个专家小组都是在 ISO 领导下的专家小组, 其小组成员也有很大的交叠。JPEG 的目标是专门集中于静止图像压缩, MPEG 的目标是针对活动图像的数据压缩, 但是静止图像与活动图像之间有密切关系。一个视频序列图像可以看作为独立编码的静止图像序列, 只是以视频速率顺序地显示。

MPEG 专家小组的研究内容, 不仅仅限制于数字视频压缩, 音频及音频和视频的同步问题都不能脱离视频压缩独立进行。MPEG-1 视频是面向比特率大约为 1.5Mbit/s 的视频信号的压缩, MPEG-1 音频是面向每通道速率为 64Kbit/s、128Kbit/s 和 192Kbit/s 的数字音频信号的压缩, MPEG-1 的最终目标还得解决数字视频和数字音频等多样压缩数据流的复合和同步的问题。所以, MPEG-1 是将数字视频信号和与其相伴随的音频信号在一个可以接受的质量下, 能被压缩到比特率约 1.5Mbit/s 的一个 MPEG 单一比特流。

MPEG 专家小组承担制定了可用于数字存储介质上的视频及其关联音频的国际标准, 这些国际标准简称为 MPEG 系列标准。数字存储介质的概念包括传统的存储设备、CD-ROM、DVD、数字音频磁带 (DAT)、磁带设备、温彻斯特硬盘、可写光盘以及电信通道, 如综合服务网 (ISDN) 和局域网等。

MPEG 标准的制定过程中竞争异常激烈,它综合了学术界和产业界的最新研究成果,并且为达到目标做了大量测试工作。不仅考虑新技术、新概念、高质量,同时在实现方案和本之间尽量折衷。仅 MPEG 视频竞争方案就有 17 个公司或学术机构提出建议,有 14 个建议接受分析和测量。

综上所述,多媒体运动图像和伴音的数据压缩编码标准,即 MPEG-1 标准,实际上包括三个部分,MPEG 视频、MPEG 音频和 MPEG 系统。本节的重点放在 MPEG 视频和音频压缩技术上。

## 2. MPEG-1 视频

### (1) MPEG-1 视频压缩的特点

MPEG 视频压缩技术,为满足应用需要,需具有以下特点。

1) 随机存取。随机存取是存储媒介上视频信息必不可少的特性。随机存取要求能在被压缩的视频比特流中间进行存取,并且能在限定的时间内对视频的任一帧进行解码。随机存取意味着存在可随机存取的单元,即某段信息编码的结果仅与该段自身的信息有关。在质量不下降的前提下,随机存取时间大约可达 0.5s。

2) 快速正向/逆向搜索。根据存储媒介的特点,对压缩数据流可进行扫描(可借助于应用规定的目录结构),也可以利用合适的存取点来显示所选择的图像,以实现正向快速搜索和逆向快速搜索。

3) 逆向重播。交互式的应用有时需要视频信号能够逆向重播,但并非所有的应用都需要在逆向重播时保持完好的画面质量。

4) 视听同步。视频信号应该准确地与相关的音频信号相同步。如果音频和视频信号分别由两个稍有差别的时钟产生,那么应该提供一个机制使这两个信号能持久地重新同步。同步特性是由 MPEG 小组提出的,MPEG 小组定义一个用于多音频、视频信号同步和合成的工具或手段。

5) 容错性。大多数数字存储介质和通信并不是不产生错误的,所以希望有一个合适的信道编码方案能适用于多种应用,并且要求这种编码方案对残存的未被校正的误差有强的鲁棒性,这样即使是在有误差的情况下,也能避免编码失败。

6) 编码/解码延迟。在视频电话的应用中,必须能够保证系统的延迟时间低于 150ms,以便保证这种面对面进行对话的应用质量要求。出版应用中,可以允许一个较长的延时,这种情况要求编、解码延时不超过 1s。传输质量和延迟在一个相当的范围内是可以折衷考虑的,因此压缩算法应在可接受的延迟范围内可充分地被执行。延迟时间被看作为一个阈值参数设定。

除以上所述的特点之外,还要求视频压缩技术具有可编辑性和灵活格式,运用计算机视频窗口技术以支持各种格式,允许各种光栅尺寸(视频屏幕的宽、高)和帧速率等。同时要求编码方案的实时完成,解码器尽可能用少量的芯片实现,以控制成本不致过高。

### (2) MPEG-1 视频压缩策略

MPEG-1 视频压缩技术是针对运动图像的数据压缩技术。为了提高压缩比,帧内图像数据压缩和帧间图像数据压缩技术必须同时使用。帧内压缩算法与 JPEG 压缩算法大致相同,采用基于 DCT 的变换编码技术,用以减少空域冗余信息。帧间压缩算法采用预测法和插补法,预测法有因果预测器(纯粹的预测编码)和非因果预测,即插补编码。预测误差可再通

过 DCT 变换编码处理，进一步压缩。帧间编码技术可减少时间轴方向的冗余信息。

1) 去时域冗余。由于 MPEG 对视频信号作随机存取的重要要求和通过帧间运动补偿可有效地压缩数据位数，MPEG 采用了三种类型的图像：帧内图 (Intra pictures, I)，预测图 (Predicted Pictures, P) 和插补图，即双向预测图 (Bidirectional Prediction, B)。帧内图可提供随机存取的存取位置，但压缩比不大；帧间插补可减少时域的冗余信息。帧间预测编码时要用到先前 (过去) 的图 (帧内图或预测图)，当前的预测图通常又作为后面 (将来) 的预测图的参考值；双向预测图的数据压缩效果最显著，但是它在预测时需要先前和后续的信息，另外，双向预测图不能作为其他图的预测参考图。帧内图 (I) 和预测图 (P) 及双向预测图 (B) 沿时间轴上的顺序排列如图 4-25 所示。

I B B P B B P B B P B B P B B I B B P B B P B B P B B P B B

图 4-25

MPEG 中这些图的组织结构是十分灵活的，它们的组合可由应用规定的参数决定，如随机存取和编码延迟等。

(a) 运动补偿。运动补偿是减少帧序列冗余信息的有效办法。运动补偿是基于  $16 \times 16$  子块的算法，每个子块可作为一个二维的运动矢量处理。运动补偿实际上是一种广义的预测技术，它适用于单纯性预测 (因果预测) 和非因果预测 (插补)。运动补偿预测是以子块 ( $16 \times 16$ ) 为预测单元，把当前子块认为是先前某一时刻图像子块的位移，位移的内容包括运动方向和运动幅度。所以运动补偿预测是用先前 (过去) 的局部图像来预测当前的局部图像， $16 \times 16$  的运动矢量块是预测误差，它必须进行编码、传送，供解码时恢复图像用。

运动补偿中的非因果性预测，即插补编码是基于时间轴上的多分辨率技术，是对时间轴 (帧序列方向) 方向上低分辨率的子信号进行编码。比如通常仅对帧率为  $1/2$  (15 帧/秒) 或帧率为  $1/3$  (10 帧/秒) 的低分辨率图像进行编码，然后作图像插值及附加校正，最后得到满分辨率的图像信号。插值法重建满分辨率图像信号的方法是把校正信息加到前面和后面参考图像组合而成的。

运动补偿插补编码也叫双向预测编码。通过双向预测编码可以获得一个高的压缩比。一个电视图像的帧序列中，不能全部是插补图 B，B 图必须由参考图进行插补，参考图可以是帧内图 (I) 或预测图 (P)，B 图不能作参考图。在两个参考图之间出现双向预测图 B 的频率是可选择的。当增加参考图之间 B 图的数目时，将会减少 B 图与参考图之间的相关性。B 图数目的选择与被编码的图像景物有依赖性，对大多数景物来说，参考图以大约  $1/10s$  的间隔隔开较为合适。

由于 I 图、P 图、B 图三者之间存在因果关系，如第 4 帧的 P 图是由第 1 帧的 I 图预测；第 1 帧 I 图第 4 帧 P 图共同预测出它们之间的双向预测 B 图，所以接受端解码器的输入 (发送端编码器的输出)，不能按照时间的顺序，而是按照以下的排列顺序：

I P B B P B B P B B P B B I B B ……

这种帧图排列顺序完全符合解码需要。解码器的输出又恢复为图 4-30 编码器输入顺序显示。

(b) 运动表示。MPEG 标准中，运动补偿估算是基于  $16 \times 16$  的块为单元表示的。这样的补偿单元称为宏块。宏块有不同的类型。比如在双向预测图 (B) 的每个  $16 \times 16$  的宏块，

可以是帧内型的、前向预测型的、后向预测型的或者是平均型的。对于一个给定的宏块，其预测器的表达式取决于参考图（前向和后向）和运动矢量。

不同区域宏块的运动矢量可有不同的选择，运动矢量的选择范围是基于帧间图像的时间分辨率和块内图像的时间分辨率以及帧序列图像的性质而选定。当两个  $16 \times 16$  宏块所包含的画面内容在传送中完全静止不动，那么宏块的运动矢量为零（宏块坐标没有改变）。

对于每个  $16 \times 16$  宏块的运动信息与其相邻块之间可作不同的编码处理。采用宏块运动补偿方法可减少电视图像帧间完整图像传送帧数，去除冗余信息，获取高压缩比和良好重建图像质量的压缩效果。

(c) 运动估算。运动的估算涉及到从视频序列中抽取运动信息所使用的一整套技术。MPEG 标准说明了怎样表示运动信息，根据运动补偿的类型：前向预测、后向预测和前后向预测，每个  $16 \times 16$  的宏块中可包含有一个或两个运动矢量，然而 MPEG 标准并没有说明运动矢量的求取方法。但是由于基于块的运动表示算法，按照尽量减小匹配误差的方法来获得运动矢量。这个匹配误差可由一个表示该块与每个预测的候选块之间的不匹配程度的代价函数来测量。设  $M_i$  是当前帧图像  $I_C$  中的一个宏块， $v$  是相对于参考图  $I_r$  的位移量，那么由最佳匹配获得的运动矢量表达式由式 4-32 表示。

$$v_i = \min_{\bar{X} \in M_i} \sum D[I_C(\bar{X}) - I_r(\bar{X} + \bar{v})] \quad (4-32)$$

$v \in V$  ( $V$  是一个  $v$  变量的集合)

其中运动矢量的可能搜索范围和匹配误差函数  $D$  的选择可在实现过程完成。穷举搜索对全部可能的运动矢量都作了考虑，结果会好，但是这个结果是以很大的计算复杂性为代价而得来的，所以在实现过程中，必须对运动矢量场的质量与运动估算过程的计算复杂性折衷考虑。

2) 去空域冗余。电视图像的帧内图像和预测误差信号都有很高的空域冗余信息。可用于减少空域冗余信息的技术很多，MPEG 优先考虑了基于块的技术。在基于块的空间冗余技术领域，变换编码技术和矢量量化编码技术是两种可选用的方法。离散余弦变换 (DCT) 编码有明确的优点和相对简单的实现方法，由 DCT 技术与视觉加权标量量化及行程编码和熵编码技术是被优先考虑的变换编码技术。基于 DCT 的变换编码技术在 JPEG 标准中已作过介绍，这里只简单归纳如下：

MPEG 标准用 DCT 技术进行帧内图像的数据压缩编码，与 JPEG 标准对静止图像的压缩编码和 CCITT H.261 标准中可视电话压缩编码处理方法是相同的。在这三个国际标准中都采用离散余弦变换编码方法。DCT 变换编码方法归纳起来可分为离散余弦变换 (DCT)、对变换系数进行量化（包括量化、Z 字扫描、行程编码）及熵编码三个阶段。

离散余弦变换是把一个  $8 \times 8$  空间窗口（块）的图像采样数据或者是预测误差数据作离散余弦正交变换，得出 64 个变换系数。

DCT 系数的量化是一步关键的操作，因为量化器结合行程编码使大部分数据得以压缩。通过量化器的量化操作，使编码器就能使它的输出与给定的位速率匹配。

(a) 视觉加权量化。量化误差的主观感觉随 DCT 系数的频率可有很大的变化，利用这一特性可对高频系数作比较粗的量化。精确的量化矩阵依赖于许多外部参数，诸如图像的显示特性，观察距离和源图像中的噪声数量，因此对某种应用或者甚至对一个单独的序



列设计一个专用的量化矩阵是合理的。一个特别的矩阵可作为编码环境和压缩的视频数据一起存储。

(b) 帧内块和非帧内块量化的比较。对于来自帧内块的信号的量化不同于由预测或插补得来的信号的量化。帧内编码的块包含所有频率的能量，如果量化太粗的话，很有可能产生块效应。另外，预测误差类型的块主要包含高频，可作更粗的量化处理。假设编码过程可以精确地预测低频，那么预测误差信号的低频分量一定是很小的。假如不是这样，在编码时就要采用帧内块类型，帧内块类型与差分编码块类型的差别导致使用两种不同的量化器结构，虽然两种量化器都是接近均匀的量化器（有一个固定的步长），但它们在零附近的特性是不同的。帧内块量化器没有死区（即量化为零值的区域比步长要小，而帧外量化器有一个大的死区）。

(c) 可调量化器。并不是所有的空间信息都能使人眼视觉系统产生同等的感觉，特别是对于那些信号变化梯度平稳的块，如果有一个非常小的误差，人眼就会觉察到块的边界（块效应或称假轮廓），而对信号变化剧烈（包含边界）的块，视觉对误差的敏感察觉被掩盖。为了适应块之间信号的不均匀性，可在块与块的基础上对量化器步长进行调节。这个机制也可用于对特定的比特率提供非常平滑的自适应调整（速率控制）。

为了进一步提高 DCT 固有的压缩性和减小运动信息对整个比特率的影响，使用可变长度的码字进行编码（即变长码）。对 DCT 系数，使用一个类似哈夫曼的表对相应于数对{行程，幅值}的符号进行编码。为了避免出现太长的符号，使用一个换码符后面跟随一个固定长度的码字。

(3) MPEG-1 视频的分层结构

MPEG-1 视频图像数据流是一个分层结构，目的是把比特流中逻辑上独立的实体分开，防止语意模糊，并减轻解码过程的负担。对分层的要求是支持通用性、灵活性和有效性。MPEG 标准的通用性可以用 MPEG 比特流来更好地说明。通用性的含义是使 MPEG 标准的语法规则可满足不同的应用要求。比如存储在计算机硬盘上的视频信号的随机存取和可编辑性，随机存取和可编辑性需要许多存储点，适时间间隔的图像组（例如 6 帧图，1/5s），并以固定数量的位数进行编码，使可编辑性成为可能。编码单元是一组图，它的编码只用组内的参考图数据。对于有噪声信道上的传送，在信道上残留未校正的误差，为提高鲁棒性（Robustness），预测器经常复位，帧内和预测图像被分割成许多片段，另外为了支持在比特流中间的“调准”，要经常对视频序列的编码内容进行重复。MPEG 标准的灵活性可通过视频序列头上所定义的许多参数来说明。虽然 MPEG 标准是针对比特率约为 1.5Mbit/s，分辨率约为 360 像素/行，但更高的分辨率和更高的比特率也是可行的。MPEG 标准的有效性是 MPEG 压缩算法需要对附加信息，如位移域、量化器步长、预测器或插值类型等提供有效的管理。

MPEG 视频比特流分层结构如表 4-13 所示。共包括六层，每一层支持一个确定的函数，或者是一个信号处理函数（DCT，运动补偿），或者是一个逻辑函数（同步，随机存取点）等。

MPEG 语法把 MPEG 比特流定义为一个

表 4-13 MPEG 视频比特流语法的六个层次

图像序列层（随机存取单元：上下文）
图像组层（随机存取单元：视频编码）
图像层（基本编码单元）
宏块片层（重同步单元）
宏块层（运动补偿单元）
块层（DCT 单元）

符合语法的二进制数字序列。另外，比特流必须满足用一个合适大小的缓冲区来进行解码的要求。在解码器的输入端设置一个尺寸适当的缓冲区，不能要求缓冲区的尺寸过分庞大，使比特率和缓冲大小匹配（既不溢出，又不浪费）便可。保证缓冲区大小给出了在视频缓冲区校验器环境内对比特流进行解码所必须的最小缓冲区的尺寸。

### 3. MPEG-1 音频

MPEG-1 音频编码过程如下：输入的音频抽样被读入编码器；映射器建立经滤波的输入音频数据流的子带抽样表示，如在层 1 或层 2 是子带抽样，在层 3 则是经变换的子带抽样；心理声学模型建立一组控制量化和编码的数据；各子带系数经过量化和编码，再加上其他一些附加信息；最后形成已编码的数据流。

有 4 种不同的编码模式：单声道、双声道、立体声和联合立体声。

根据应用需求，可以使用不同层次的编码系统，编码器的复杂性和性能也随之升高。

（1）层 1 包括将数字音频变成 32 个子带的基本映射。将数据格式化成固定分段的块。决定自适应位分配的心理声学模型。利用块压扩和格式化的量化器。理论上，层 1 编/解码最小延迟为 19ms。

（2）层 2 提供了位分配、缩放因子和抽样的附加编码。使用了不同的帧格式。这一层最小编/解码延迟为 35ms。

（3）层 3 采用混合带通滤波器来提高频率分辨率。它增加了差值量化（非均匀）、自适应分段和量化值的熵编码。这一层最小编/解码延迟为 59ms。

符合 MPEG-1 标准的编码器能产生一个合乎 MPEG-1 标准的语法结构的 MPEG-1 比特流的编码器。MPEG-1 标准为在多媒体存储媒介上视频信号规定了一个语法以及与此语法相关的含义，一个解码器能够对 MPEG-1 比特流进行解码，产生的结果在解码过程所规定的可接受的范围内的解码器称之为 MPEG-1 解码器。MPEG-1 标准只规定了比特流语法和解码过程，用户可很好地利用这个语法的灵活性来设计非常高质量的编码器和非常低成本的解码器。编码器的设计中的一些重要参数，如运动估算、自适应量化和位速率控制等可由用户自由确定。

速率约为 1.2Mbit/s 的用 MPEG-1 算法压缩的视频图像的质量相当于 VHS（家用视频系统）记录质量。空间分辨率限制为每视频扫描行 360 个像素，并且在源编码器端的视频信号为 30 帧/秒，非隔行扫描。对大多数原始图像内容，可得到无人工痕迹的图像质量。

MPEG-1 中视频序列参数的灵活性使之产生了许多特性，诸如支持很宽范围的空间和时间分辨率，能使用很大范围的比特率。但是保证使用 MPEG-1 标准的设备之间的相互操作性更为重要，不能强迫设备厂商再建造一个额外设计的系统。由于这个原因，MPEG-1 定义了参数空间的一个特殊的子集，它给出了 MPEG-1 标准主要目标内的一个较为合理的折衷。MPEG-1 标准的主要目标是要使编码的视频信号比特率约为 1.5Mbit/s。用表中的参数定义了一个约束参数比特流。

MPEG-1 标准是 VCD 工业标准的核心，现在已经走入千家万户；利用 MPEG-1 音频第三层的 MP3 音乐格式也倍受青睐。

### 4.5.3 MPEG-2 标准

同 MPEG-1 标准一样，MPEG-2 标准也包括系统、视频和音频等部分内容；它克服并解决了 MPEG-1 不能满足日益增长的多媒体技术、数字电视技术对分辨率和传输率等方面的

技术要求的缺陷。

MPEG-2 标准的系统功能是将一个或更多的音频、视频或其他的基本数据流合成单个或多个数据流,以适应存储和传送。符合 MPEG-2 标准的编码数据流可以在一个很宽的恢复和接收条件下进行同步解码。MPEG-2 系统支持五项基本功能:解码时多压缩流的同步;将多个压缩流交织成单个的数据流;解码时缓冲器初始化;缓冲区管理;时间识别。MPEG-2 标准的压缩编码系统是将视频和音频编码算法结合起来而开发的。系统编码可有两种方法,其编码输出包括程序流和传送流两种定义流。程序流和 MPEG-1 系统定义的流相似;而传送流是一种用来传送和保存程序的编码数据或其数据的数据流。

MPEG-2 视频体系要求必须保证与 MPEG-1 视频体系向下兼容,并同时应力求满足数字在存储媒体、会议电视/可视电话、数字电视、高清晰度电视(HDTV)、广播、通信、网络等应用领域对多媒体视频、音频通用编码方法日益增长的新需要。如分辨率要求有低( $352 \times 288$  像素)、中( $720 \times 480$  像素)、次高( $1440 \times 1080$  像素)、高( $1920 \times 1080$  像素)不同档次;压缩编码方法也要求从简单到复杂有不同等级。

MPEG-2 国际标准详述了数字存储媒体和数字视频通信中的图像信息的编码描述和解码过程。MPEG-2 标准支持固定比特率传送、可变比特率传送、随机访问、信道跨越、分级解码、比特流编辑以及一些特殊功能,如快进播放、快退播放、慢动作、暂停和画面凝固等。MPEG-2 视频标准与 MPEG-1 视频向前兼容并与 EDTV, HDTV, SDTV 格式向上或向下兼容。

MPEG-2 标准作为计算机可处理的数据格式主要应用于数字存储媒体、视频广播和通信。存储媒体可以直接与 MPEG-2 解码器相连,或者通过总线、局域网、电信通信线路等通信手段与其相连。所以符合 MPEG-2 标准的数据,可以在现在或未来的网络上传送、接收和在现在或未来的广播信道上传播。

MPEG-2 视频具有以下特色:

### (1) 框架和级别

框架是 MPEG-2 标准中定义的语法的子集。级别是 MPEG-2 标准规范的一个特定框架中的参数所取值的集合。一个框架可以包含一个或多个级别。MPEG-2 标准为了实现一个完整的语法体系,通过框架和级别的方式来约定有限数目的语法子集。给定某框架所规定的语法范围后,比特流参数的取值仍然要影响编码和解码过程,所以在每个框架中定义了“级别”。级别是一个对比特流各参数进行限定的集合。例如, MPEG-2 图像规范中,以帧宽(水平像素数)、帧高(垂直像素数)和帧率(帧数/秒)的乘积作为约束分辨率等级参数:低级[ $352 \times 288 \times 29.79$ ],基本级[ $720 \times 480 \times 29.79/720 \times 480 \times 25$ ],次高级[ $1440 \times 1080 \times 30/1440 \times 1080 \times 25$ ],高级[ $1920 \times 1080 \times 30/1920 \times 1080 \times 25$ ]。框架和级别提供了一种定义 MPEG-2 规范的语法和语义的子集的手段。框架和级别限定之后,解码器的设计和解码校验就可针对限定的框架在限定的级别中进行。同时以框架和级别的形式定义规范为不同的应用领域之间的数据交换提供了方便和可行性。

### (2) 视频压缩编码的数据结构

视频压缩编码的视频数据结构是分层的比特流结构,第一层称为基本层,基本层可以独立被解码,其他层称之为增强层,增强层的解码依赖于基本层。基本层的结构与 MPEG-1 相一致。视频数据结构的编码比特流包括视频序列层、图像组块层、宏块层和块层。视频序列处于最高层。视频序列从视频序列头开始,后面紧接着是一系列数据单元。当序列头中除了包括有序列

头函数[Sequence—header( )]之外还有序列扩展函数[Sequence—extension( )]时, MPEG-1 规范不再适用, 只适用于 MPEG-2 规范。在视频序列头中包括编码扫描方式(逐行/隔行, 帧图/场图); 图类型, (I 图, P 图, B 图), I 图使用自身图进行编码, P 图是使用先前的 I 图或 P 图信息预测编码, B 图是由过去或将来的 I 图或 P 图双向预测编码图。MPEG-2 没有 D 图, D 图是变换系数的直流分量(DC 系数), 代表能量分布的图, 而 MPEG-1 中有 D 图。

为了提供随机访问的功能, 在编码比特流中可有重复序列头出现, 重复序列头只可以在 I 图或 P 图前面出现, 不能在 B 图前面出现。I 图用以解决视频序列的随机访问问题, 如节目重播、快进播放或快退播放等。

视频序列层之下是图像组块层, 组块是由宏块构成, 一个组块可由多个宏块组成。宏块结构可有三种格式:

- 4:2:0 格式。一个宏块由六个块组成。其中包括四个亮度(Y)块, 两个色差块(Cb 块和 Cr 块)。
- 4:2:2 格式。4:2:2 格式由八个块组成。其中包括四个亮度(Y)块, 四个色差块(两个 Cb 块和两个 Cr 块)。
- 4:4:4 格式。这种宏块结构由 12 个块组成。其中包括四个亮度(Y)块, 八个色差块(四个 Cb 块和四个 Cr 块)。

对帧图使用 DCT 编码的宏块结构和场图使用 DCT 编码的宏块结构是不一样的。帧 DCT 编码中, 每个块由两场扫描行交替组成; 场 DCT 编码中, 每个块仅由两场中之一的场扫描行组成。

组成宏块的块是 DCT 变换的最基本单元。块尺寸为  $8 \times 8$  (以图像像素为单位)。像素即图像采样样本, 是画面显示的最小单元, 其尺寸大小取决于分辨率。每个像素携带有亮度 Y 和色差信号 Cb、Cr 信息。MPEG-2 视频采样格式也有三种格式: 4:2:0、4:2:2 和 4:4:4。

### (3) 语义规则

MPEG-2 对视频比特流的语义规则也作了具体规定, 制定了更高层语法结构的语义规范。其高层比特流组织中, 不带扩展功能的数据流与 MPEG-1 规范一致。

MPEG-2 视频解码器的芯片级产品已被世界上许多著名的大公司纷纷推出, 是工业标准 DVD 的核心标准。

MPEG-2 音频与 MPEG-1 差别不大, 不过它支持 5+1 声道。

要了解 MPEG-2 标准的详细情况, 请参阅钟玉琢、乔秉新和祁卫译著的《运动图像及其拼音通用编码国际标准——MPEG-2》, 清华大学出版社 1997 年 6 月出版。

## 4.5.4 MPEG-4 标准

### 1. 引言

MPEG-4 在扩展性上具有很好的灵活性, 可进行时域和空域的扩展。这在 MPEG-2 中也有些体现, 但它并不是很突出。而在 MPEG-4 中, 可根据现场带宽和误码率的客观条件, 在时域或空域进行扩展, 时域扩展是在带宽允许时在基本层之上的增强层中增加帧率, 在带宽窄时可在基本层中减少帧率, 以达到充分利用带宽, 使图像质量更好; 在空域扩展时是指对基本层中的图进行采样插值, 增加或减少空间分辨率。

为了支持前面提到的各种功能: 高效压缩、基于内容交互(操作、编辑、访问等)以及

基于内容分级扩展（空域分级、时域分级），必然要求 MPEG-4 要以基于内容的方式表示视频数据。因此，MPEG-4 中引入视频物体 VO（Video Object）和视频物体平面 VOP（Video Object Plane）等概念来实现基于内容的表示。VO 的构成依赖于具体应用和系统实际所处的环境，对于低要求应用情况下，VO 可以是一个矩形帧（即传统 MPEG-1、H.263 中的矩形帧），从而与原来的标准兼容；对于基于内容的表示要求较高的应用来说，VO 可能是场景中的某一物体或某一层面，VO 也可能是计算机产生的二维、三维图像等。当 VO 被定义为场景中截取出来的不同物体时，每个 VO 有三类信息来描述：运动信息、形状信息、纹理信息。所以 MPEG-4 标准的视频编码是针对这三种信息的编码技术。

基于内容的视频编码过程可由三步完成：

- 1) VO 的形成先要从原始视频流中分割出 VO。
  - 2) 对各 VO 分别独立编码，即对不同 VO 的三类信息（运动信息、形状信息、纹理信息）分别编码，分配不同码字。
  - 3) 将各个 VO 的码流复合成一个符合 MPEG-IV 标准的比特流。
- 在编码和复合阶段可以加入用户的交互控制或由智能化算法进行控制。

## 2. VOP 的编码

MPEG-4 标准提供灵活的框架和开放的工具集，它通过工具集和句法描述语言（MSDL）不同的组合，支持功能的不同组合。

下面对基于 VOP（Video Object Plane）的编码进行介绍。

VO（Video Object）是场景中的某个物体，它由时间上连续的帧画面序列构成。VOP 是某一时刻某一帧画面的 VO，VOP 编码即针对某一时刻该帧画面 VO 的形状、运动、纹理等三类信息进行编码。

### （1）形状编码

一个从场景中截取出的 VOP 如图 4-26 所示。

由图可见，VOP 是一个不规则的形状。表示 VOP 的开头可用二值图表示，或者用灰度图描述。通常以 8 位表示灰度，可有 256 级（0~255）灰度分层，二值图形状只需 1 位表示（0，1）。标准约定“0”表示非 VOP 区域（背景），“1”表示 VOP 区域，对于用灰度图表示的 VOP 形状，“0”表示非 VOP 区域，“1~255”表示 VOP 区域。以灰度表示 VOP 形状时，物体与背景的边界轮廓线比二值图表示要柔和。

MPEG-4 标准形状编码方法是用位图法，VOP 被一个边框“框住”，边框长、宽均为 16 的整数倍，同时保证边框最小。位图表示法实际上就是一个边框矩阵，矩阵元素为 0~255（或 0，1），编码变为对这个矩阵的编码。矩阵被分成  $16 \times 16$  的“形状块”（如图 4-26 所示），边界信息包含在块中。位图法不是 VOP 形状编码的唯一方法，比如将 VOP 用梯度图表示，形状边界被抽出，用边界跟踪方法进行编码也很方便，可以断定，将来的标准中将会引入其他基于几何轮廓的编码技术。

### （2）运动估计和运动补偿

MPEG-4 标准中的 VOP 运动估计和运动补偿与以前的压缩标准（MPEG-1，H.263 等）一样。

类似于以前的压缩标准（MPEG-1，H.263 等）的三种帧格式：I-帧图、P-帧图、B-帧图，MPEG-4 中 VOP 也有三种相应的帧格式：I-VOP、P-VOP、B-VOP（如图 4-27 所示），

以表示运动补偿类型的不同。

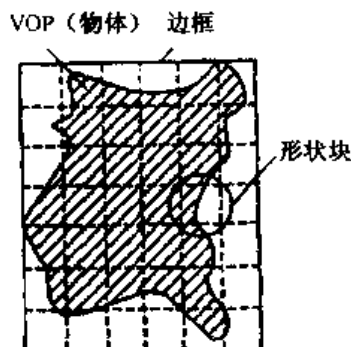


图 4-26 VOP 形状编码

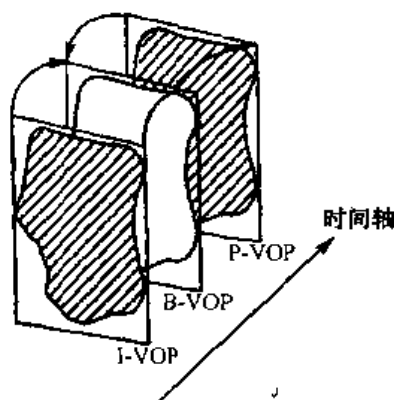


图 4-27 VOP 帧编码类型

VOP 也如形状编码那样，外加了边框，边框分成  $16 \times 16$  的宏块，宏块由  $8 \times 8$  的块构成，运动估计和补偿即可基于宏块，也可基于块。

MPEG-4 的运动估计和补偿与 H.263 非常接近，采用了 H.263 中的“半像素搜索” (Half Pixel Searching) 技术和“重叠补偿” (Over Lapped Motion Compensation) 技术。为了使 H.263 的运动估计和补偿算法能适用于任意形状的 VOP 区域，引入了所谓“重复填充” (Repetitive Padding) 和“修改的块 (多边形) 匹配” (Modified Block (polygon) Matching) 技术。

### (3) 纹理编码

纹理信息用两种，可能是内部编码的 I-VOP 的像素值，也可能是帧间编码的 P-VOP、B-VOP 的运动估计残差值。为了达到简单、高性能、容错性好的目的，仍采用基于分块的纹理编码。VOP 边框仍被分成  $16 \times 16$  的宏块，宏块由  $8 \times 8$  的块构成，图 4-28 示出 VOP 的基于宏块的纹理编码图。DCT 变换基于  $8 \times 8$ ，仍有三种情况：VOP 外、边框内的块：不编码；VOP 内的块：传统 DCT 方法编码；部分在 VOP 内、部分在 VOP 外的块：先用“重复填充”方法将该块在 VOP 外的部分进行填充（对于残差块只须填零），再用 DCT 编码。这样是为了增加块内数据的空域相关性，从而利于 DCT 变换和量化去除块内的空域冗余。

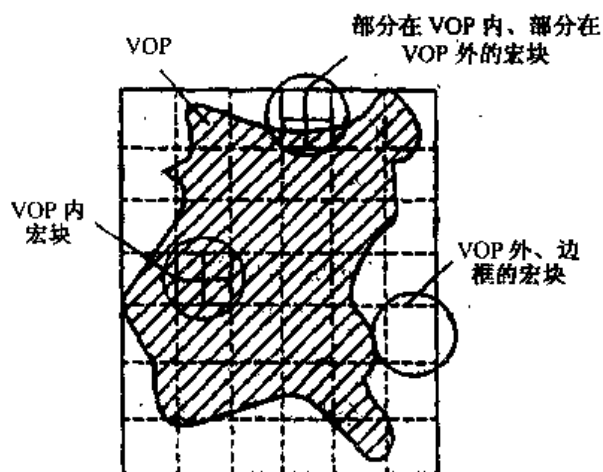


图 4-28 任意形状 VOP 的基于宏块的纹理编码

DCT 系数要经量化、Z-扫描、游程及哈夫曼熵编码。

### (4) 分级扩展编码

MPEG-4 的数据结构如图 4-29 所示。

一个完整的视频序列由几个视频段 VS 构成，每个 VS 由一个或多个 VO 构成，每个 VO 又由一个或多个视频物体层 VOL 构成。每个 VOL 代表一个层次（基本层、增强层），每个层表示某一种分辨率。在每个层中，都有时间上连续的一系列 VOP。

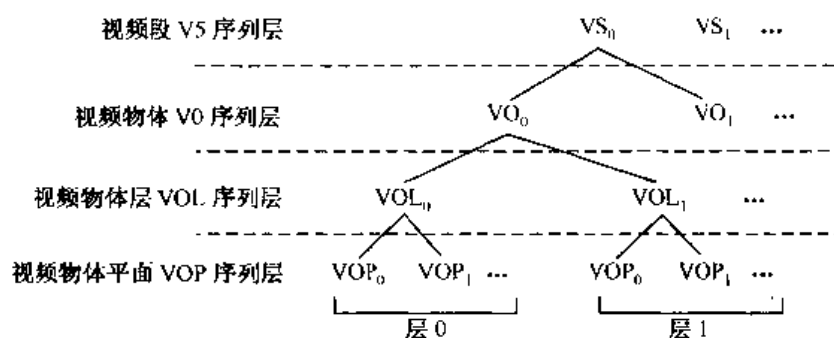


图 4-29 MPEG-4 的数据结构类分级图

利用 VOL 结构实现空域分级扩展和时域分级扩展是 MPEG-4 的一个重要功能。要实现分级扩展至少要有两个 VOL，一个基本层，一个增强层。空域分级是用增强层来增加基本层的空域分辨率；时域分级就是由增强层来增加基本层中感兴趣区域的时域分辨率——帧率。

### 3. 视频流的数据结构

MPEG-4 视频流中包括以下几种数据：视频数据、静态纹理数据、2-D 网格数据和人脸活动参数数据等。这些结构化数据称为对象，对象和它们的属性按分层形式组织，以便支持码流可扩展性和对象可扩展性。

编码的视频数据是由一种被称为层的视频码流组成的有序集。如果只有一层，那么称这个视频流是不可扩展的；反之，如果有两层以上，那么就称这个视频流是可扩展的。这些层中有一层称为基本层，它总是可以单独解码的。其他层称为增强层，它们只能同基本层和较低的增强层一起解码。这些层的多路技术在 MPEG-4 系统流 (ISO/IEC 14496-1) 中讨论。基本层可以用其他标准编码，而增强层的编码将遵循 MPEG-4 视频流 (ISO/IEC 14496-2) 语法。一般地说，MPEG-4 视频流可以看作一个语法层次，其中的一个语法结构包含一个或更多的从属结构。

视觉纹理（参看这里的静态纹理编码）是纹理在各种条件下（以二维和三维合成场景的交互为典型）传输和复制时为维持高视觉质量而设计的。静态纹理编码提供了亮度、颜色和形状的多层表现。这支持了纹理的累进传输，可以实现分辨率和质量的扩展性编码，对网络下载十分有利。

一些可以被映射成网格的纹理对象单独编码。编码的网格数据由一个不可扩展的码流组成。这个码流定义了 2D 网格对象的结构和运动。

编码的人脸活动参数数据也由一个不可扩展的码流组成，它为解码器定义了人脸模型。人脸活动数据按一个标准格式组织，它包括一个可下载的人脸模型和为远程操纵这个模型而使用的控制参数。人脸的形状、纹理和表示一般用码流中的面部定义参数 (Facial Definition Parameter FDP) 集和面部活动参数 (Facial Animation Parameter FAP) 集控制。在初始化之前，人脸对象是一个自然表示的通用人脸，在不断从码流中接收面部活动参数时逐渐生成面部活动。如果码流中有面部定义参数，那么就可以生成特定形状和外表的人脸。

#### (1) 视觉对象序列

视觉对象序列是视频流的最高语法结构。一个视觉对象序列由视觉对象序列起始码 (visual\_object\_sequence\_start\_code) 开始，后面跟着一个或多个视觉对象。视觉对象序列由视觉对象序列结束码 (visual\_object\_sequence\_end\_code) 指示其结束。

## (2) 视觉对象

一个视觉对象由视觉对象起始码 (visual\_object\_start\_code) 开始, 后面接着是框架和级标识和视觉对象标识, 然后是视频对象、静态纹理对象、网格对象或人脸对象。

## (3) 视频对象

一个视频对象由视频对象起始码 (video\_object\_start\_code) 开始, 接着是一个或多个视频对象层。

1) 帧。一帧由三个整数矩阵组成: 一个亮度矩阵 (Y) 和两个色差矩阵 (Cb 和 Cr)。

2) 视频对象平面 (Video Object Plane——VOP)。一个重建的 VOP 由解码一个编码的 VOP 获得, 而编码的 VOP 来源于一个累进帧或交织帧。

3) VOP 类型。有四中类型的 VOP, 它们用不同的方法编码:

- 内部 VOP (I-VOP) 只用自己的信息编码。
- 单向预测 VOP (P-VOP) 根据它前面的 VOP 利用运动补偿技术来编码。
- 双向预测 VOP (B-VOP) 根据它前面和后面的 VOP 利用运动补偿技术来编码。
- 全景 VOP (S-VOP) 用来编码 Sprite 对象。

4) I-VOP 和 VOP 组。I-VOP 是对序列进行随机访问的标识。一些需要随机访问的操作 (例如快进和快退) 常常要频繁地访问 I-VOP。I-VOP 还被用在场景剪切和运动补偿失效时。VOP 组 (GOV) 是由一个 I-VOP 开始的若干 VOP 的组合, 并用一个头标志来指示解码器。

5) 格式。在这里两个色差 (Cb 和 Cr) 矩阵无论在水平还是在垂直方向都可以是亮度 (Y) 矩阵尺寸的一半 (亮度矩阵要保证采样为偶数)。亮度和色差的采样位置由图 4-30 所示。对交织 VOP 在垂直和时间上采样的两种变化如图 4-31 和 4-32 所示, 图 4-33 显示了累进编码时在垂直和时间上的采样位置。

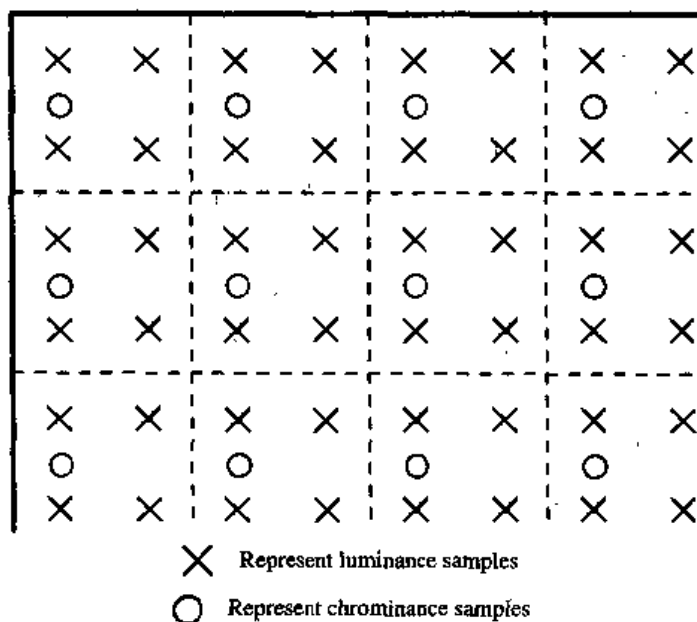


图 4-30 4:2:0 格式时亮度和色差采样位置

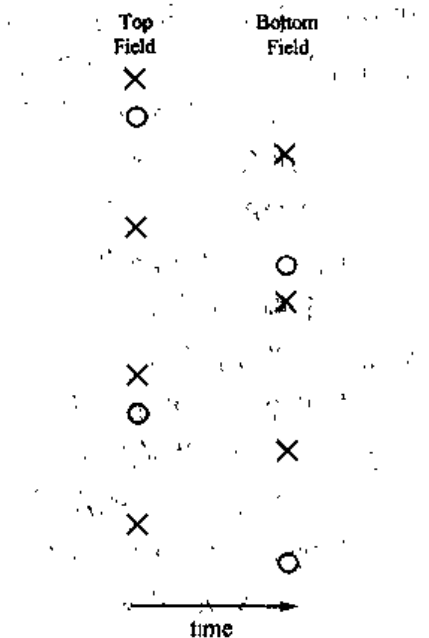


图 4-31 在交织帧中 top\_field\_first=1 时垂直和时间的采样位置



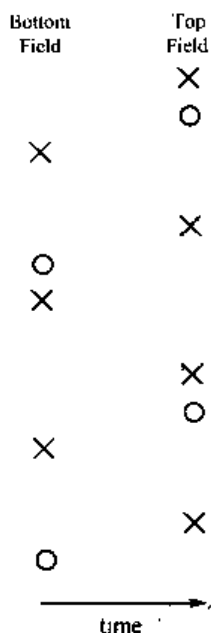


图 4-32 在交织帧中  $\text{top\_field\_first}=0$  时垂直和时间的采样位置

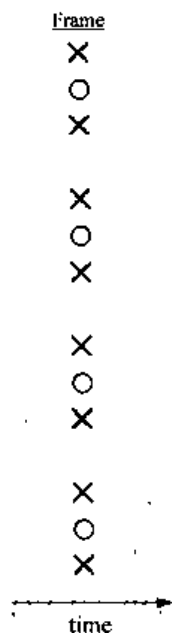


图 4-33 在累进帧中垂直和时间的采样位置

每一个 VOP 的形状用二值  $\alpha$  平面表示时，指示形状的矩形总是和包围亮度 VOP 的矩形尺寸相当。在这个矩形中，4:2:0 格式时亮度和色差像素的采样位置如图 4-30 所示。在累进情形下，亮度像素的每一个  $2 \times 2$  子块对应一个色差像素；而在交织情形下，亮度像素的每一个  $2 \times 2$  子块对应同一场中的一个色差像素。

为了在两个色差平面中进行填充过程，必须为色差平面生成和它相适应的  $\alpha$  平面来指示其形状（长和宽均约为亮度  $\alpha$  平面的一半）。因此，当使用不可扩展的形状编码时，色差  $\alpha$  平面就必须用亮度  $\alpha$  平面生成，这是一个亚采样过程。

对于亮度  $\alpha$  平面（不分累进和交织情形）的每一个  $2 \times 2$  子块，只要四个采样值中有一个是 255，那么色差  $\alpha$  平面中的采样值就设定为 255。

6) VOP 重排。当一个视频对象层包含 B-VOP 时，连续 B-VOP 的数目是变化的，而且是巨大的。但是编码的第一个 VOP 不可以是 B-VOP。

一个视频对象层可以不含 P-VOP，也可以不含 I-VOP，在这种情形中视频对象层的开始和内部必须提供有效的随机访问和错误恢复手段。

在码流中 VOP 的编码顺序（也称为解码顺序）是解码器用来重建视频对象时需要的 VOP 顺序。而解码过程输出的重建 VOP 的顺序（也称为显示顺序）一般不和解码顺序相同，因此解码器必须用一定的规则在解码过程中对 VOP 进行重排。

当视频对象层中不含 B-VOP 时，解码顺序和显示顺序是相同的。

而当 B-VOP 出现在视频对象层中时，重排按以下规则进行：

- 如果在解码顺序中当前的 VOP 是 B-VOP，那么输出的 VOP 就是这个 B-VOP。
- 如果解码顺序中当前的 VOP 是 I-VOP 或 P-VOP，那么输出的 VOP 是前一个解码的 I-VOP 或 P-VOP。当然，如果是在视频对象层的开始，前面没有已解码的 VOP，那么没有输出。

下面是一个从视频对象层开始的 VOP 序列的例子，这里两个 P-VOP 或相邻的 I-VOP

和 P-VOP 之间都有两个 B-VOP。VOP ‘1I’ 作为 VOP ‘4P’ 的参考 VOP，而 VOP ‘4P’ 和 ‘1I’ 一起作为 VOP ‘2B’ 和 ‘3B’ 的参考 VOP。因此在码流中 VOP 的编码顺序是 ‘1I’、‘4P’、‘2B’、‘3B’，而在解码器解码时的输出为 ‘1I’、‘2B’、‘3B’、‘4P’。编码器的输入顺序如下：

1	2	3	4	5	6	7	8	9	10	11	12	13
I	B	B	P	B	B	P	B	B	I	B	B	P

编码器的输出顺序，也就是解码的输入顺序如下：

1	4	2	3	7	5	6	10	8	9	13	11	12
I	P	B	B	P	B	B	I	B	B	P	B	B

解码器的输出顺序如下：

1	2	3	4	5	6	7	8	9	10	11	12	13
I	B	B	P	B	B	P	B	B	I	B	B	P

7) 宏块。一个宏块包含有亮度成分和相应的色差成分。宏块这个术语既可以描述未编码的源数据和解码后的重建数据，也可以来描述编码的数据。一个略过宏块是不携带任何信息的宏块。目前宏块中只有一种色差格式（称为 4:2:0）。宏块中块的顺序如下面所述：

一个 4:2:0 的宏块包含 6 个块，其中 4 个 Y 块、1 个 Cb 块和 1 个 Cr 块。

VOP 中宏块的组织如下所述：

对累进 VOP 情形，交织标志（在 VOP 头中）被设为 0，这时 VOP 中宏块的组织被称为帧组织格式，如图 4-34 所示。这种情形下，帧 DCT 编码被使用。

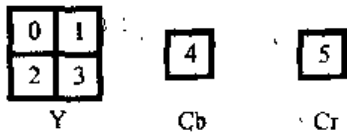


图 4-34 4:2:0 宏块结构

对交织 VOP 情形，交织标志被置为 1。这时 VOP 中的宏块组织既可以采用帧组织方式，也可以采用场组织方式，同时也相应地使用帧 DCT 编码和场 DCT 编码。

- 在帧 DCT 编码情形，每一个亮度块被逐行放在一起形成，如图 4-35 所示。

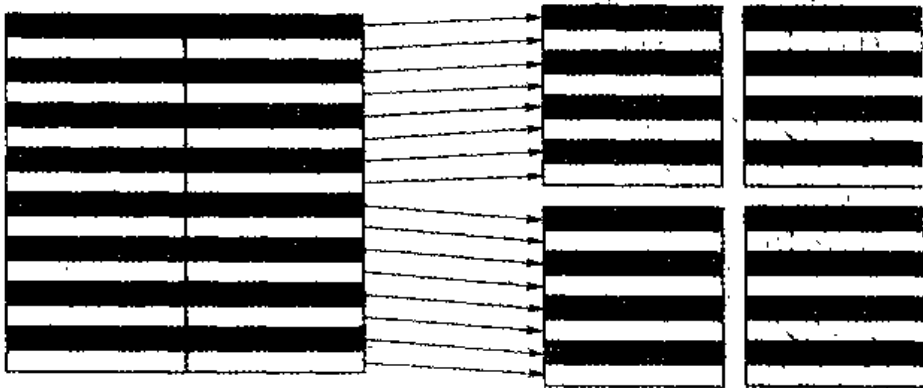


图 4-35 帧 DCT 编码时亮度宏块的组织

- 在场 DCT 编码情形，每一个亮度块被隔行放在一起形成，如图 4-36 所示。

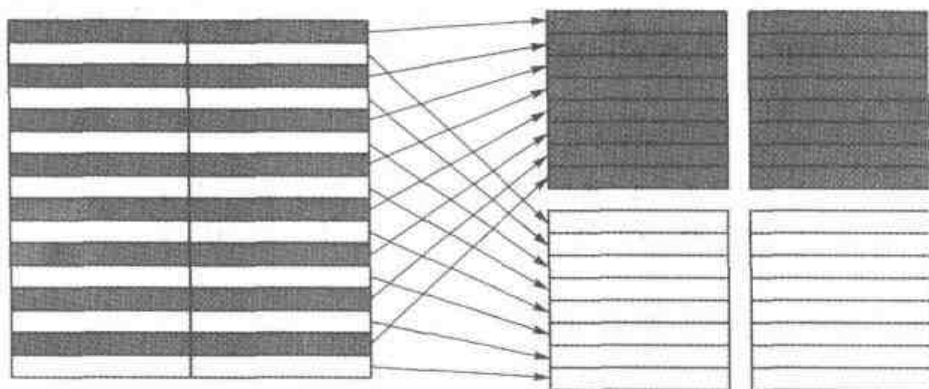


图 4-36 场 DCT 编码时亮度宏块的组织

在编码色差块时，只能用帧 DCT 编码。这时需要注意，这些色差块可以使用基于场的预测，因为可能要在  $8 \times 4$  的区域内进行预测（在半采样滤波后）。

8) 块。块这个术语既可以描述未编码的源数据和解码后的重建数据，又可以描述 DCT 系数，还可以指编码的数据元素。这里的块是 8 行  $\times$  8 列的。

#### (4) 网格对象

1) 概述。一个 2D 的三角形网格涉及到将 2D 视觉对象平面分割成小三角形。这些三角形小片的顶点被称为结点。结点之间的直线段被称为边。如果两个三角形共用一条边，我们称它们相邻。

一个动态 2D 网格由一个 2D 三角形网格的时间序列组成，这些网格必须有相同的拓扑结构，但是它们的顶点位置彼此不同。因此一个动态 2D 网格可以用 2D 网格的几何学和网格结点的运动向量（这些运动向量指明了视频对象序列中网格结点的运动情况）来描述。一个 2D 动态网格可以利用纹理映射创建 2D 动画，例如将视频对象平面映射成连续的 2D 网格。

一个网格可以带暗示结构，用来描述统一的或具有 Delaunay 拓扑结构的 2D 网格。在这两种情形下，初始化网格的拓扑结构可以不编码（因为已经暗中定义了），不过初始化网格的结点位置还需要编码。注意在这两种情形，网格被限制的很简单，它只能包含一些简单连接，而不能有洞。

一个网格对象表现为 2D 网格的几何形状和运动。网格对象由一个或多个网格对象平面（在一定时间内符合 2D 三角形网格的一个实例）组成。网格对象的一个例子如图 4-37。

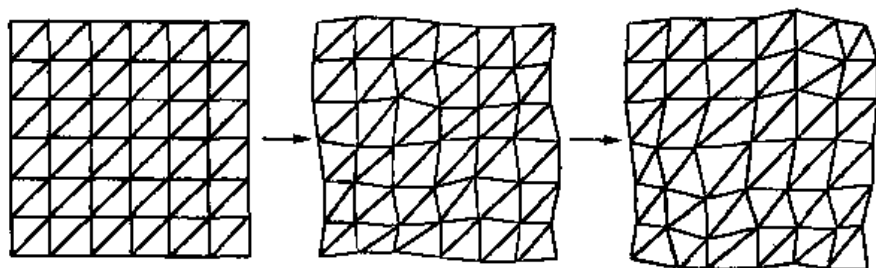


图 4-37 单一三角形的网格对象

一网格对象平面的序列表现了由小片组成的物体的变形情况，它可以利用视频对象平面和静态纹理对象产生合成动画对象。视频对象平面的三角形小片可以用符合三角形网格元素的方式扭曲。网格元素的运动可以用网格结点在时间上的移动来说明。

网格对象的语法和语义只适合网格的几何学和运动，如果视频对象要用作动画需要单独编码，那么使网格对象变成视频对象平面的扭曲或纹理映射由场景合成的上下文控制。此外，这个语法不能清楚地说明其他网格属性的编码，例如颜色和纹理。

2) 网格对象平面。有两种类型的网格对象平面，它们用不同的方式编码。

内部编码的网格对象平面，编码单一的 2D 网格的几何形状，它既可以是统一型，也可以是 Delaunay 类型。当网格是统一型时，网格的几何形状用小参数集编码；而网格是 Delaunay 类型时，网格的几何形状用结点和边的位置来编码。三角形网格结构用编码信息隐含说明。

预测编码的网格对象平面根据过去的参考网格对象平面利用时间预测来编码。预测编码网格的三角形结构和参考网格是相同的，只是结点位置有所变化。结点的移位表明了网格的运动，这用当前网格对象平面相对参考网格对象平面的结点运动向量来说明。

网格结点的位置和视频对象和静态纹理对象中位置概念一致。网格对象结点位置和运动向量也是半像素精度的。

(5) 人脸对象

从概念上说，人脸对象由一个场景图（在人脸码流中描述）的结点集合组成。人脸的形状、纹理和表示一般由包含人脸定义参数（FDP）集和人脸活动参数（FAP）集的码流来控制。在解码之前，人脸对象包含一个带自然表示的一般人脸。这个人脸可能已经被展现，可以马上从码流中接收 FAP 来生成人脸的活动，例如表示和语言等。如果收到 FDP，就利用 FDP 中确定的形状和纹理将一般的人脸变换成特定的人脸。一个完整的人脸模型通过 FDP 集用类似向场景图中插入人脸结点的方法生成。

FDP 集和 FAP 集用来定义人脸的形状和纹理，也就是重构人脸表示、表情和语言的活动。恰当地说，FAP 可以在对不同的人脸模型时对表示和语言取得相当好的结果，而不需要重新初始化和校准模型。FDP 可以在设置时定义人脸形状和纹理的精度。如果在初始化时使用 FDP，可以对特定人脸特征得到更精细的运动。在 FAP 转化中利用音位和书签，可以通过 TTS 系统得到 FAP 来控制人脸模型。从音位到 FAP 的转化并未标准化，它假设每一个解码器都有带缺省参数的缺省人脸模型。因此，在初始化阶段并不需要建立人脸活动，解码器端在初始化阶段只要将人脸用户化。

1) 人脸对象码流的结构。一个人脸对象由一组人脸对象平面的时间序列组成。这可以由图 4-38 描述。

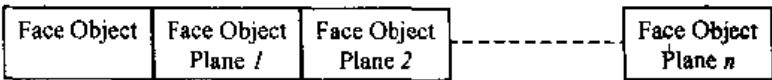


图 4-38 人脸对象码流的结构

一个人脸对象表现了 MPEG-4 标准(ISO/IEC 14496)的场景图的一个节点。一个 MPEG-4 标准的场景根据一些时空关系当作视听对象来理解，这些场景图是分层表现的。

作为选择，人脸对象也可以由人脸对象平面组的时间序列来组成，其中一个人脸对象平

面组是由 16 个人脸对象平面的时间序列组成的。

当这种可选择的人脸对象码流结构被采用时，人脸编码用的是基于 DCT 的人脸编码。否则，人脸编码用的是基于帧的人脸对象编码。

2) 人脸活动参数集。FAP 是基于最小人脸活动的研究而得出的，和肌肉活动密切相关。它们描述了基本人脸活动的一个完全集，因此可以对绝大部分自然人脸进行表示。夸张地说，它可以定义常人不可能作出的人脸动作，这可以用来生成幽默讽刺画。

FAP 集包含两种高级参数：Viseme 和 Expression。Viseme 是对音素的视觉相关。Viseme 参数允许 Viseme 翻译其他参数没有的表示，并且增强其他参数的表示结果。只有静态的 Viseme 包括在标准集中，附加的 Viseme 可能被加到将来的标准扩展中。类似地，Expression 参数定义了高级的人脸表示，它通过纹理描述来定义。为了使人脸活动简单，常聚在一起表示人脸的 FAP 被编成一个 FAP 组。这些表示参数是描述人脸活动的非常有效的手段。

3) 人脸活动参数单元。所有包含运动信息的参数都在人脸活动参数单元(Facial Animation Parameter Units FAPU)中表示。这些单元用统一的方式对人脸模型进行描述而定义，以便对表情语言等生成合理的结果。它们符合一些人脸关键特征之间的距离因素，作为特征点之间的距离来定义。一些测量单元如表 4-14 所示，在这里符号 3.1.y 描述了特征点 3.1 的 Y 坐标，可以参看图 4-39 所示。

表 4-14 人脸活动参数单元

	描 述	FAPU 值
$IRISD0 = 3.1.y - 3.3.y = 3.2.y - 3.4.y$	自然人脸的虹膜直径	$IRISD = IRISD0 / 1024$
$ES0 = 3.5.x - 3.6.x$	两眼距离	$ES = ES0 / 1024$
$ENS0 = 3.5.y - 9.15.y$	眼睛和鼻子的距离	$ENS = ENS0 / 1024$
$MNS0 = 9.15.y - 2.2.y$	嘴和鼻子的距离	$MNS = MNS0 / 1024$
$MW0 = 8.3.x - 8.4.x$	嘴的宽度	$MW = MW0 / 1024$
AU	角度单元	$10^{-3} \text{ rad}$

4) 自然人脸的描述。在码流开始，人脸被假设为一个自然姿势；当 FAP 全为 0 时也显示一个自然人脸。所有 FAP 参数表示了自然人脸特征点位置的移动。自然人脸有以下定义：

- 坐标系统是右手坐标系，头轴平行于世界轴。
- 凝视方向时 Z 轴。
- 所有人脸肌肉时松弛的。
- 眼皮接触虹膜。
- 瞳孔是 IRISD0 的三分之一。
- 两片嘴唇接触，唇线水平，并且两嘴角等高。
- 嘴紧闭，上下齿接触。
- 舌头平坦，舌尖接触上下齿。

5) 人脸定义参数集。FDP 来用户化私有的人脸模型，

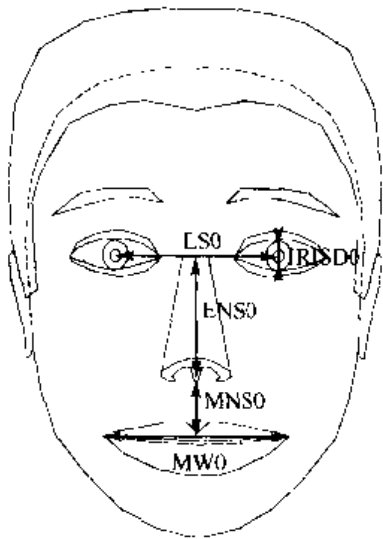


图 4-39 人脸活动参数单元

以便让解码器得到特殊的人脸模型（当然还要附加活动信息）。FDP 参数通常每一个镜头传输一遍，然后才是压缩的 FAP 参数。然而如果解码器没有收到 FDP 参数，要保证仍然可以利用 FAPU 中的 FAP 参数解释人脸。这样可以在广播和视频会议中只使用最小的操作。FDP 集在 BIFS 中说明（参看 MPEG-4 系统流 ISO/IEC 14496-1）。FDP 节点定义了接收器使用的人脸模型，有两种意见被支持：

- 校准信息是下载的，以便使接收器的私有人脸信息可以随意地用人脸特征点、3D 网格和纹理配置。
- 人脸模型和人脸活动参数定义一起下载，由接收器替换人脸模型。

#### 4. MPEG-4 音频流

##### (1) 概述

ISO/IEC 14496-3 (MPEG-4 音频) 是一种新的音频标准，它综合了许多种不同类型的音频编码，例如带合成声音的语音编码、高质量低码率的传输编码、带音乐的话音编码、复杂的音轨编码和一些交互以及虚拟现实的内容。通过单独标准化一些成熟的编码工具，形成了完美且灵活的音频编码和合成的框架，MPEG-4 音频标准的开发者创建了交互式数字化音频世界的新技术。

MPEG-4 和从前由 ISO/IEC 或其他组织创建的音频标准不同，它不是针对单一的应用，例如实时技术或高质量音频压缩。确切地说，MPEG-4 音频是可以应用到所有需要使用先进的语音压缩、合成、控制和回放等应用领域的标准。MPEG-4 音频中有许多领域的编码工具，但是 MPEG-4 音频不止是这些部分的总和。这里描述的工具和 MPEG-4 标准的其余部分结合起来，可以实现基于对象的音频编码、交互式表示和动态音频跟踪等。

因为工具的一个子集用来覆盖多种应用的需要，所以互用性是 MPEG-4 音频系统的自然特征。使用特殊编码的系统，例如使用 MPEG-4 语音编码工具集的实时语音通信系统，可以方便地和其他系统共享数据和开发工具，甚至可以和使用相同工具的不同领域的系统互用数据和工具，例如使用 MPEG-4 语音编码的语音检索系统。一个能解码 MPEG-4 音频的主要框架的多媒体终端有能力覆盖所有现在和将来可用的音频功能。

##### (2) MPEG-4 音频中的新概念

MPEG-4 中的许多新概念和从前的 MPEG 音频标准相比有许多不同。为了方便熟悉 MPEG-1、MPEG-2 和 MPEG-AAC 的读者，我们简单叙述一下这些不同之处。

1) MPEG-4 没有传输标准。在所有 MPEG-4 的音频和视频编码工具中，编码标准在创建包含压缩数据的单元序列后就结束了。MPEG-4 系统 (ISO 14496-1) 描述了如何将各个单独编码的对象转换为一个包含多个子码流的码流。这里没有为码流在信道上传输的标准机制，因为可以使用 MPEG-4 技术的应用十分广泛，所以没有一个简单的结论适用多种传输模式。

2) MPEG-4 音频能实现低码率编码。先前的 MPEG 音频标准主要着眼于需要的各种码率下的高质量音频的透明或几乎透明的编码。MPEG-4 为这个目的提供了新的改进工具，这些标准化工具还可以用来在适合 Internet、数字广播或其他带宽有限的传输中使用的低码流语音的传输中。

3) MPEG-4 是有多种工具的基于对象的编码标准。从前的 MPEG 音频标准提供了单一的工具集，通过不同的配置使这些工具适用于多种应用。MPEG-4 提供了几种工具集，它们之间没有特别的联系。MPEG-4 音频的框架描述了如何将这些工具使用在各种不同应用中。

此外,在从前的 MPEG 标准中只能传输单一内容;而在 MPEG-4 中则相反,声音轨迹的概念很灵活。多种工具用来传输几个音频对象,当共同使用多种工具时,一个音频合成系统用来从音频码流中创建一个声音轨迹。

4) MPEG-4 提供了声音合成能力。在自然语音编码中,用服务器压缩存在的声音,传输它,在接收端解压缩。这样的编码类型是许多存在的声音压缩标准的主题。MPEG-4 还标准化了合成声音(包括语音和音乐)描述。

和从前的 MPEG 一样, MPEG-4 没有标准化音频编码方法。这样,作者在创建音频码流时就可以各尽所长,使用最好的方法。现在,如何将自然声音转换为合成音频或多对象描述是一个开放的问题。

### (3) MPEG-4 音频的性质

#### 1) 性能综述:

MPEG-4 音频工具可以组织为几个种类:

- 语音工具 (Speech tool)。关于合成的和自然的语音的传输和解码。
- 音频工具 (Audio tool)。关于记录的音乐和音轨的传输和解码。
- 综成工具 (Synthesis tool)。关于合成的音乐和其他声音的低码率描述、传输和在终端的合成。
- 合成工具 (Composition tool)。关于基于对象的编码、交互的功能和视听对象的同步。
- 可扩展性工具 (Scalability tool)。关于在不同码率下传输码流的准则。

#### 2) MPEG-4 语音编码工具:

(a) 在 MPEG-4 中提供了两种语音编码工具。自然语音编码工具允许为电话、个人通信和监视应用进行人类语音的压缩、传输和解码。综合语音工具提供了一个文语转换综合系统的接口,使用合成语音可以提供甚低码率操作,建立和低码率视频电话会议应用的联系。下面将逐个讨论这些工具。

(b) 自然语音编码。MPEG-4 语音编码工具集包括了码率范围从 2 到 24Kbit/s 的自然语音的压缩和解码。当允许变码率编码时,那么也可以支持低于 2Kbit/s 的编码。一个是参数化语音编码算法 HVXC (Harmonic Vector eXcitation Coding),另一个是 CELP (Code Excited Linear Prediction) 编码技术。MPEG-4 语音编码器针对的应用从移动卫星通信到 Internet 技术、多媒体包装和语音数据库。

使用码率扩展技术, MPEG-4 的 HVXC 可以在码率 2.0~4.0Kbit/s 之间工作。它还可以在更低的可变码率下工作,典型为 1.2~1.7Kbit/s。HVXC 提供了在 8kHz 采样率下使用 100~3800 Hz 频带的通信质量。HVXC 还允许在解码过程中独立地改变速度和音高,这对于快速访问语音数据库是一个强大的功能。

MPEG-4 的 CELP 是众所周知的具有新功能的编码算法。传统的 CELP 器提供单一码率的压缩,并且为了指定应用可以进行优化。压缩只是 MPEG-4 CELP 提供的一个功能, MPEG-4 还允许在多种应用使用一个基本的编码器。它在码率和带宽上提供了扩展性,也就是可以在任意码率下产生码流。MPEG-4 CELP 编码器支持两种采样率: 8 和 16 kHz。和 8 kHz 相对应的带宽是 100~3800 Hz, 而和 16 kHz 相对应的带宽是 50~7000 Hz。

(c) 文语转换能力现在变成一个相当普通的媒体,它在各种多媒体应用领域中起着重

要的作用。例如，通过使用 TTS (Text To Speech) 功能，带叙述的多媒体内容可以在不录制自然语音的情况下轻松地创建。在 MPEG-4 之前，多媒体内容提供者没有办法容易地为未知的 TTS 系统给出说明。在 MPEG-4 中，将对于 TTS 系统的单一的公共的接口标准化。

MPEG-4 TTS 包（混合/多级可扩展 TTS 接口）可以考虑为传统 TTS 框架的一个超集。这个扩展的 TTS 接口可以利用从除了输入文字的自然语音产生的韵律信息，从而产生更高质量的合成语音。这个接口和它的码流格式在附加信息项中是高度可扩展的，例如，如果韵律信息的某些参数不可用，解码器可以通过某些规则产生丢失的参数。语音合成和文字到因素转换的标准算法不在 MPEG-4 中指定，但是要符合 MPEG-4 TTS 接口的目标，解码器要按照用户需要使用提供的所有信息。

作为文语合成系统的接口，MPEG-4 指定了韵律信息、人脸活动参数和其他活动参数的联合编码方法。使用这个技术，一个码流可以用来控制文语接口和人脸活动视觉对象解码器。这样扩展 TTS 的功能范围从传统的 TTS 到自然语音合成，它的应用领域从简单的 TTS 到利用 TTS 进行和音频表示和运动图像配音。

3) MPEG-4 通用语音编码工具。MPEG-4 标准化了码率从 6~64Kbit/s（这是对每一个声道而言，也就是对立体声最大可到 128Kbit/s，对 5 声道的环绕立体声可到 320Kbit/s）的自然语音的编码。使用 MPEG-2 AAC 标准 (ISO/IEC 13818-7) 可以提供通用的高质量的压缩，在 MPEG-4 中做了一定的改进。

MPEG-4 通用语音编码技术使用了感觉滤波器组、一个成熟的屏蔽模型、噪声形状技术、声道合并、无噪声编码和为给定质量下提供最大压缩的位分配。MPEG 开发心理声学编码标准已经有 10 年了，MPEG-4 通用语音编码继续了这一传统。

对码率从 6~64Kbit/s 的情形，MPEG-4 标准化了码流语法和在工具集中项目的解码过程。提供的不同工具可以让作者依赖信号的特征选择使用的工具获得高质量的结果。

4) MPEG-4 音频综合工具。提供了通用音频合成能力的 MPEG-4 工具集称作 MPEG-4 结构化音频。MPEG-4 结构化音频 (SA) 为合成声音描述提供了非常通用的能力，也为解码器端的综合声音提供了准则。

和综合的特殊方法不同，SA 为描述综合方法定义了灵活的语言。这个技术为内容作者提供了两个有利条件。第一，可用的合成技术集并不限于标准创建者指定的，MPEG-4 结构化语音可以使用当前或未来的各种合成方法。第二，MPEG-4 中结构化描述的合成语音准则是精确标准化的，所以 SA 编码器保证了它在各个终端上以相同的方式工作。

合成语音通过工具模块传输，这些工具模块可以在一定范围的控制下创建音频信号。一个工具是信号处理的小网络，它按照一些算法控制了语音的参数生成。几个不同的工具可以在单个结构化语音中传输和使用。

通过提供和 MIDI 制造组织下载声音级别 2 (DLS-2) 标准（它是结构化语音标准的标准参考）的互用性可以完成在采样合成中使用的声音采样（也叫做波表）的有效的传输。通过使用 DLS-2 格式，波表合成的简单而通俗的技术可以在 MPEG-4 结构化音轨中使用。为了进一步允许和存在内容和创作工具的互用性，通俗的 MIDI (Musical Instrument Digital Interface) 控制可以代替或加在控制合成的 SASL 的使用范围中。

通过包含和 MIDI 标准的兼容性，MPEG-4 结构化音频标准表示了当前的声音合成描述（基于 MIDI 的波表合成）的技术和未来的技术（一般目的的算术合成）的统一性。因而指



定的标准不仅解决了其低码率的编码问题，也解决了虚拟环境、视频游戏、交互式音乐、卡拉 OK 系统和其他一些应用。

5) MPEG-4 语音合成工具。音频合成工具和视频合成工具一样，在 MPEG-4 系统标准 (ISO/IEC 14496-1) 指定，这里提供了一个综述。

音频合成用来控制单独的音频对象，它也是创建单一音轨的混合技术。这和多路（包括音乐工具、话音和音效）混合中记录音轨的过程是类似的，然后把它们合成一个或两个声道。在 MPEG-4 中，可以传输多声道的混合，其中每一个音频源可以使用不同的编码工具，不过混合的说明也要在码流中传输。在收到多个音频对象时，分别独立地把它解码，但不要把它们播放给用户；然后使用混合说明生成给定对象的单一音轨，再把最终的音轨播放给用户。

6) MPEG-4 语音扩展工具。MPEG-4 中许多码流类型都是按某种方式可扩展的。下面讨论标准中几种扩展的类型。

码率可扩展性允许将一个码流分割成低码率的码流，将它们合并在一起又可以得到更高质量的信号。码流的切分可以发生在传输中，也可以发生在解码器中。扩展性可以在每一种自然音频编码方案中使用，也可以在不同的自然音频编码方案中合并。

带宽可扩展性是码率扩展性的一种特殊情形，因为表示频谱的某部分的码流部分可以在传输或解码时丢弃。这个技术可以用在 CELP 语音编码器中，这里扩展层将窄带的基本层编码器转换成宽带的语音编码器。在频域工作的通用音频编码工具也提供了对不同编码层的灵活的带宽控制。

编码器复杂性的可扩展性允许编码器在不同复杂度的情形下产生合理而有意义的码流。其中的一个例子是宽带 CELP 编码器的高质量和低复杂性的扩展性，它允许在降低编码器复杂度和优化编码质量之间作出选择。

解码器复杂性的可扩展性允许解码器在不同的复杂性级别上解码给定码流。解码器复杂性的可扩展性的一个子类型是完美的退化 (Graceful Degradation)，这里解码器可以动态地监视可用资源，当资源受到限制时可以降低解码复杂性（也降低了音频质量）。结构化音频解码器允许这类扩展性，内容创建者可以对不同的声音提供几种合成算法，依赖可使用的资源解码器可以选择其中一种。

## 4.5.5 MPEG-7 标准

### 1. 概述

如今，越来越多的声像信息以数字形式存储和传输，这为人们更灵活地使用这些信息提供了可能性。但随之而来的问题是，随着网络上信息爆炸性的增长，获取到我们感兴趣的信息的难度却越来越大。传统的基于关键字或文件名的检索方法显然不适于数据量庞大、又不具有天然结构特征的声像数据，因此近些年来多媒体研究的一个热点是声像数据的基于内容的检索，例如“从这段新闻片中找出有克林顿的镜头”这种形式的检索。实现这种基于内容检索的一个关键性的步骤是要定义一种描述声像信息内容的格式，而这与声像信息的存储形式（编码）又是密切相关的。国际标准化组织 ISO/IEC 下辖的运动图像专家组（简称 MPEG）注意到了这方面的需求和潜在的应用市场，在推出影响极大的 MPEG-1、MPEG-2 之后，尚未完成 MPEG-4 的最后定稿，便开始着手制定专门支持多媒体信息基于内容检索的编码方案：MPEG-7。

MPEG-7 作为 MPEG 家族中的一个新成员, 正式名称叫作“多媒体内容描述接口”(Multimedia Content Description Interface), 它将为各种类型的多媒体信息规定一种标准化的描述, 这种描述与多媒体信息的内容本身一起, 支持用户对其感兴趣的各类“资料”的快速、有效地检索。

各种“资料”包括: 静止图像、图形、音频、动态视频, 以及如何将这些元素组合在一切的合成信息。

这种标准化的描述可以加到任何类型的多媒体资料上, 不管多媒体资料的表示格式如何, 或是什么压缩形式, 加上了这种标准化描述的多媒体数据就可以被索引和检索了。

各种类型的信息的标准化描述可以分成一些语义上的层次。以视频资料为例: 较低层次的描述就是颜色、形状、纹理、空间结构等信息; 而最高层次的语义描述信息可以是: “画面中有一只棕色的狗在左边, 而一个蓝色的球落在右边”。也可以有介于上面两种层次之间的中层语义描述信息。同样的内容根据不同的应用领域要求可以携带不同类型的描述信息。

以下一些应用领域将从 MPEG-7 标准的制定中获益:

- 数字化图书馆(图像分类目录, 音乐字典, …)。
- 多媒体目录服务。
- 广播式媒体选择(收音机频道, 电视频道, …)。
- 多媒体编辑(个人电子新闻服务, 媒体著作)。

还有一些潜在的应用领域:

- 教育。
- 旅游信息。
- 娱乐(例如寻找游戏、卡拉 OK 节目)。
- 购物(例如寻找你喜欢的衣服)。

## 2. MPEG-7 的制定工作

在叙述 MPEG-7 的制定之前, 我们先介绍一些基本术语。

(1) 数据(Data)。MPEG-7 的数据指的是使用 MPEG-7 标准描述的视听信息, 而不管它是以何种方式存储、编码、显示、传输的。这就是说 MPEG-7 的数据可以包括视频、电影音乐、文本以及其他一些媒体, 例如 MPEG-4 流、录像带和打印的一本书。

(2) 特征(Feature)。特征是数据的与众不同的部分, 它既可以是主观的, 也可以是客观的。例如图像中的颜色和纹理、视频中的形状和运动等。

(3) 描述符(Descriptor)。描述符和一个或多个特征的表述值相联系。一个特定的特征可以有多种表示。例如颜色直方图表示颜色, 多边形定点集表示形状。

(4) 描述方案(Description Scheme)。描述方案定义了描述以及它和数据模型关系的结构和语义。一个 DS 可以看作是描述描述符之间关系的方案。

(5) 描述(Description)。描述是描述数据的实体, 它由 DS 和相关的描述符的实例组成。

(6) 编码的描述(Coded Description)。编码的描述是描述的一个表示, 它能有效的存储和传输。

(7) 描述定义语言(Description Definition Language)。描述定义语言是用来制定 DS

的语言。DDL 允许创建新的 DS。

MPEG-7 要标准化以下内容:

- (1) 一个描述方案和描述符的集合。
- (2) 一个指定描述方案的语言, 即 DDL。
- (3) 一个描述的编码策略。

注: MPEG-7 的工作重点在视听数据, 它不再为文本创建新的描述方案。

MPEG-7 不标准化特征或描述符的抽取算法, 因此特征抽取不是 MPEG-7 的标准部分。和从前的 MPEG-7 标准一样, 这样做的目的是: 第一, 可以使这些算法的新进展及时物化, 避免未来的 MPEG-7 的应用; 第二, 厂家可以在这些算法中体现自己的特色, 充分发挥自身优势。搜索引擎和数据库的组织也是 MPEG-7 的非标准部分。

和以前的 MPEG-7 standard 标准一样, MPEG-7 只标准化它的码流语法, 亦即只规定解码器的标准, 而编码器的标准不在标准之内。

- (1) 1998 年 10 月征集建议。
- (2) 1999 年 2 月评测。
- (3) 1999 年 12 月第一个委员会草案。
- (4) 2000 年 10 月委员会草案。
- (5) 2001 年 2 月最终委员会草案。
- (6) 2001 年 7 月国际标准草案。
- (7) 2001 年 9 月国际标准。

## 4.5.6 MPEG-21 标准

世界正迈进数字化、网络化、全球一体化的信息时代。信息技术将渗透到人类社会的方方面面, 其中网络技术和多媒体技术是促进信息社会全面实现的关键技术。

MPEG 曾成功地发起并制定了 MPEG-1、MPEG-2 标准, 也因此赢得了 Emmy 奖。现在 MPEG 组织也已完成了 MPEG-4 标准的 1、2、3、4 版本的标准, 2001 年 9 月完成 MPEG-7 标准的制定工作, 同时在 2001 年 12 月完成 MPEG-21 的制定工作。

MPEG-1 标准: 1988 年 5 月开始, 1992 年 11 月形成国际标准 (IS: international standard), 编号是 ISO/IEC11172, 是用于运动图像及其伴音的编码, 码率可达 1.5Mbit/s。这是世界上第一个集成视、音频编码标准, 它是第一个只定义解码器 (接收机) 的标准, 它是第一个独立于视频格式的视频编码, 第一个以软件方式制定并含软件实施的标准。现主要应用于 VCD, 其音频三层即 MP3 广泛应用于网络音乐。

MPEG-2 标准: 1990 年 6 月开始, 1994 年 11 月形成国际标准。编号是 ISO/IEC13818, 是一种通用的运动图像视、音频编码标准。码率可达 9Mbit/s。广泛应用于数字机顶盒、DVD 和数字电视。

MPEG-4 标准: 1993 年 7 月开始, 1999 年 5 月形成国际标准 (版本一)。编号是 ISO/IEC14496, 是一种基于对象的可视和音频对象编码标准, 视频码率 5Kbit/s~5Mbit/s, 音频码率 2Kbit/s~64Kbit/s。在以前系统、视频、音频、符合性测试、参考软件的基础上增强了第六部分 DMIF (多媒体传输接口)。它包容了以前的 MPEG-1 和 MPEG-2 标准, 制定了大范围的级别和框架, 可广泛应用于多种行业。

MPEG-7 标准：1997 年 7 月开始，在 2001 年 9 月达到形成国际标准。编号是 ISO/IEC15938，它不是内容的分析也不是内容的检索，更不是信息压缩编码技术，而是一种多媒体内容描述的标准，定义了描述符、描述语言和描述方案，便于对多媒体等内容的处理。可应用于数字图书馆、各种多媒体目录业务、广播媒体的选择遗迹多媒体编辑等领域。

## 1. MPEG-21 的研究目标和用户需求

### (1) 研究目标

构建于无所不在的国际通信网络如互联网基础之上的数字项消费市场已经改变了原有的物质商品交易的商业模式，可以通过网络进行数字内容的电子交易。在这种新市场中，对于多媒体内容的知识产权很难分清楚，数字视频、音频、合成图形等使消费者感到迷惑，这些内容将如何获取、如何得到？这样就需要一种新的、统一的、和谐的方法来传送不同类型的多媒体，这对于用户来说可以进行透明的多媒体信息服务。当然这只是研究问题的一个方面，还有许多其他方面的问题，如内容检索、服务质量的保证等。此外，许多个人生成了许多数据媒体来供个人使用和家庭或朋友分享，如大量的图片、音乐以及媒体贸易等共享网站。这些“内容提供者”（Content Providers）同商业内容提供者一样关心相同的事情：内容的管理、基于消费者和设备能力的内容重定位、权利的保护、非授权存取和修改保护、内容提供者和消费者个人隐私的保护等。这就要求消费者的选择要全球化。目前还没有什么标准或系统具有这种能力。

开展 MPEG-21 研究的目的是：是否需要和如何将这不同的组件（协议、标准、技术等）有机地结合在一起；讨论是否需要新的规范；如果具备了前面两个条件，如何将不同标准集成在一起。MPEG-21 的范围可以描述成是一个决定性（关键）技术的集成，这些技术可以通过访问全球网络和设备实现对多媒体资源的透明和增强的使用。其功能包括内容创建、内容产品、内容发布、内容消耗和使用、内容表示、知识产权管理和保护、内容识别与描述、财政管理、用户的隐私权、终端和网络资源抽取、事件报告等。MPEG-21 的基本框架如图 4-40 所示，包括数字项目说明（Digital Item Declaration）、内容表示（Content Representation）、数字项目的识别与描述（Digital Item Identification and Description）、内容管理与使用（Content Management and Usage）、知识产权管理与保护（Intellectual Property Management and Protection）、终端与网络（Terminals and Networks）、事件报告（Event Reporting）。虽说这七大要素存在功能上的重叠性，但它们之间的区别足以使它们标准化。

### (2) 用户需求

多媒体框架 MPEG-21 中的用户可以看作是任何与 MPEG-21 环境交互或使用 MPEG-21 数字项的实体。这些用户包括个人、消费者、团体、组织、合作者、财团、政府以及世界上所有其他的标准和自发的个人或团体。用户的界定依赖于交互中与其他用户之间的关系。从一个纯粹的技术角度看，MPEG-21 对于“内容提供者”和“消费者”没有任何区别，他们

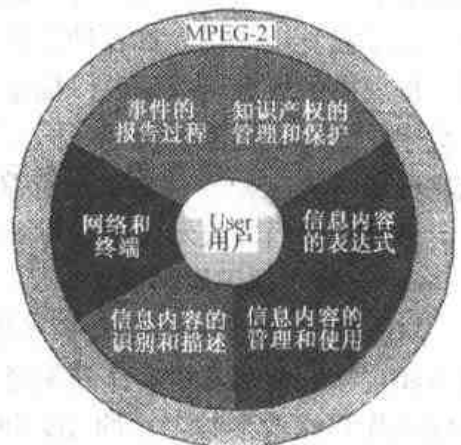


图 4-40 MPEG-21 基本框架

都是用户。由于 MPEG-21 中的试题可以有多种方法利用或使用内容，所以与 MPEG-21 交互式的所有部分都等同归类于用户。MPEG-21 中的用户特权或义务由与其他用户的交互来界定。如图 4-41 所示为 MPEG-21 中拥护定义的示意图。



图 4-41 用户定义示意图

在 MPEG-21 的框架中一个用户与另一个用户所交互的对象为数据项，一般称这些数据项为内容，这种交互包括建立内容、提供内容、存档内容、估价内容、增强和传送内容、合计内容、传送内容、联合内容、零售内容、消费内容、预定内容、调整内容、上述情况的交易以及上述情况的控制。

## 2. MPEG-21 框架中的要素

### （1）数字项说明

数字项说明的目的是建立数字项的统一的和灵活的摘要和数字项的可互操作性方案，在 MPEG-21 系统中有许多问题涉及到“数字项”，所以说对于数字项的定义应有一个具体的描述什么可以代表“项”。显然对于同一内容有许多描述方法，因此希望有一个强有力的、方便的数字项模型来表示无数种形式中的数字项的描述。如果模型产生的数字项能够不模糊的表示并且可以与模型中所定义的其他数字项互操作，则这个图形是有用的。

### （2）多媒体内容表示

MPEG-21 提供的内容表示可以通过分级和错误恢复方法有效地表示任何数据类型，自然的和合成的、或者是自然与合成组合的，等等。多媒体场景的不同元素可以单独地访问，可以同步和复用，也允许各种各样的交互式访问。显然多媒体内容是多媒体框架中的基本要素。框架中的内容可以编码、描述、存储、传送、保护、交易、消费等。尽管框架中的内容是数字化的，但还需要满足不同需求的内容数字表示，这些需求通过未压缩的数据格式是不能完成的。大家知道，如果没有编码技术，数据内容就不能应用于多种服务。在 MPEG-21 中，多媒体的内容表示可完成对 MPEG-21 基本对象的表示。

### （3）数字项识别与描述

MPEG-21 数字项识别与描述将提供如下功能：精确、可靠和独有的识别；不考虑自然、类型和尺寸的情况实现实体的无缝（seamless）识别；相关数据项的稳固和有效的识别方法；任何操作和修改下数字项的 ID 和描述都能够保证其安全性和完整性；自动处理授权交易、内容定位、内容检索和内容采集。

### （4）内容管理和使用

内容管理与使用是通过内容发布、消费的价值链，能够对内容进行建立、操作存储和再利用等。随着时间的推移，网络的内容及对内容的存取将以指数的规律增长。MPEG-21 的目的是通过各式各样的网络或设备透明的使用网络内容，现对于内容的检索、定位、缓存、存档、跟踪、发布、而且使用显得越来越重要。从这些方面考虑，内容管理应包括个性化以及用户框架（Profile）管理。MPEG-21 的内容管理与使用标准化的目的可归纳为：能够通

过内容发送和消费链建立、操作、存储、传送和使用内容的接口和协议，重点是根据用户的个性化的内容管理改善交互模型。

#### (5) 知识产权管理与保护

MPEG-21 多媒体框架应提供数字权利的管理与保护。

1) 允许用户表达他们的权利、兴趣以及各类与 MPEG-21 数字项相关的认定等，可通过大范围的网络和设备对这些权利、兴趣和认定实现稳固的和可靠的管理和保护。

2) 某种程度上能够获得、编撰、传播相关的政策、法规、合约以文化准则，从而建立针对于 MPEG-21 数字项的商业数字平台。

3) 从可能的某种程度上提供一个统一的领域管理组织和技术来管理与 MPEG-21 交互的设备、系统、应用等，提供各种商业交易的服务。目前大多数存在的电子内容是通过基本 IPMP 系统管理。现在没有什么框架可以实现这些系统的互操作性。对于消费者来说问题是不同 IPMP 系统中电子内容交流的互操作性。电子内容的权利持有者希望能够自由选择通道和技术（包括 IPMP 系统）来提供他们的内容。消费者在某些情况下希望能够自由处理他们的私人问题，包括通过匿名与内容交互。大多数存在的 IPMP 系统不能精明地处理知识产权法律的相关问题。

#### (6) 终端与网络

1) 通过屏蔽网络和终端的安装、管理和实现问题，使用户能够透明地或互操作地存取或发布高级多媒体内容。

2) 框架允许与任意用户的连接，可根据用户的需求提供网络和终端资源。

3) 上面所述意味着网络能够提供根据用户和网络的 QoS（服务质量）的内容传输。网络和终端应根据内容的要求提供内容的可分级性功能。应通过标准接口访问网络和终端资源。MPEG-21 的目标是允许通过宽范围的网络设备对多媒体资源的透明使用而不必考虑网络和终端。用户在存取内容时应提供一个明确的主题感知服务。他们应屏蔽于网络和终端的安装、管理和应用等相关问题。随着多种类型网络如有线的、无线的，GPRS、UMTS、DVB-T、-S、xDSL、LMDS、MMDS 等的到来，这种方便的使用越来越显得重要了。这就意味着高组用户的要求参数可以在网络和终端上形成透明的映射，用户可以在不管内容在网络的终端上如何传输的情况下根据 QoS 获得服务。

#### (7) 事件报告

事件报告使用户能够精确理解框架中所有可报告事件的接口和计量（metrics）。事件报告将为用户提供特定交互的执行方法，同样允许大量超范围的处理，允许其他框架和模型与 MPEG-21 实现互操作。

事件报告可为组织提供相关的及时的策略信息。包括精确的产品价值和利益信息，精确的消费价值和利益信息，精确的通道消耗和利益信息。可以将信息以正确的形式在适当的时候发给合适的人。用户可以模仿“what-if”场景做出重大决策。允许用户理解操作过程，这样可以模拟操作过程来优化效率和输出。对于用户到用户、用户与数字项、数字项与数字项的事件计算结果有利于用户分析事件。通过这些计算和报告用户可以更好地理解 MPEG-21 中的活动事件，如：交易完成与终止的比例、成员中订户的比例、订户所有的时间、消费者的平均忠实度、产品采用的比率等。

## 4.6 小结

本章详细讨论了多媒体数据压缩编码及算法的基本概念、原理以及各种编码压缩技术。这些压缩技术可以归纳为有失真压缩和无失真压缩。有失真压缩能提供较高的压缩比,但由于损失了信源的熵,压缩后的数据是无法准确无误地恢复的,无失真压缩能准确无误地恢复原信息,它只是去掉了信源的冗余部分,但却不能提供较高的压缩比。

对多媒体数据压缩各种编码技术均作了详细的讨论。如统计编码的基本原理、信息量和信息熵的概念、哈夫曼编码、算术编码、预测编码的基本原理、自适应预测编码、帧间预测编码、变换编码的基本原理及 DCT 变换等。

本章对多媒体数据压缩编码的国际标准,如静态图像压缩编码标准 JPEG 的编码解码的基本原理及实现技术,以及动态图像压缩编码标准 MPEG 的基本原理均作了很详细的分析和讨论。本章的最后对新一代的声像编码国际标准 MPEG-4、MPEG-7、MPEG-21 作了有关的介绍。

## 4.7 习题

1. 下列哪些说法是正确的?

- (1) 冗余压缩法不会减少信息量,可以原样恢复原始数据。
- (2) 冗余压缩法减少了冗余,不能原样恢复原始数据。
- (3) 冗余压缩法是有损压缩法。
- (4) 冗余压缩的压缩比一般都比较小。

- (A) (1) (3) (B) (1) (4)
- (C) (1) (3) (4) (D) 仅 (3)

2. 图像序列中的两幅相邻图像,后一幅图像与前一幅图像之间有较大的相关,这是\_\_\_\_\_。

- (A) 空间冗余 (B) 时间冗余
- (C) 信息熵冗余 (D) 视觉冗余

3. 下列哪种说法不正确?

- (A) 预测编码是一种只能针对空间冗余进行压缩的方法。
- (B) 预测编码是根据某一模型进行的。
- (C) 预测编码需将预测的误差进行存储或传输。
- (D) 预测编码中典型的压缩方法有 DPCM、ADPCM。

4. 下列哪种说法正确的?

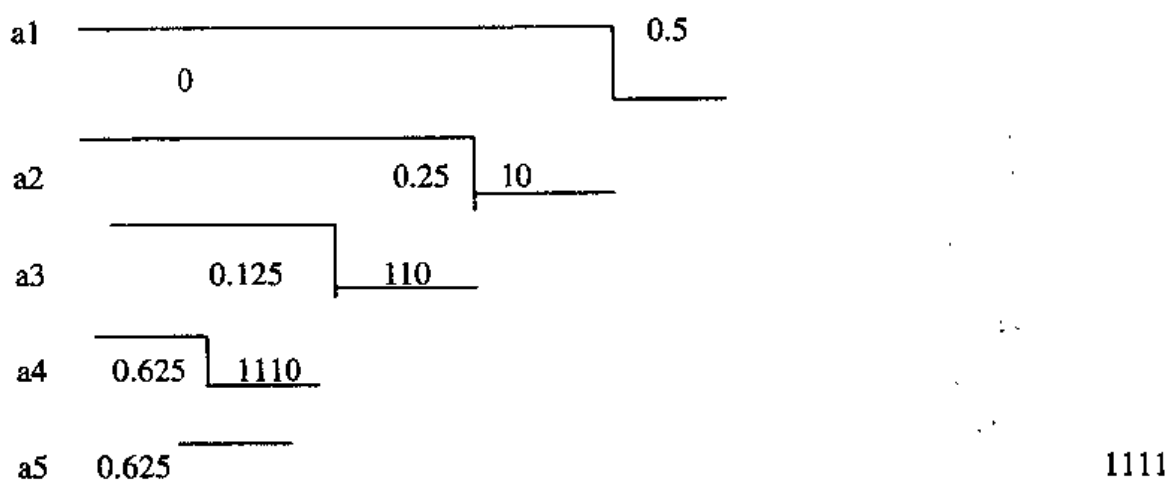
- (A) 信息量等于数据量与冗余量之和。
- (B) 信息量等于信息熵与数据量之差。
- (C) 信息量等于数据量与冗余量之差。
- (D) 信息量等于信息熵与冗余量之和。

5. P×64K 是视频通信编码标准,要支持通用中间格式 CIF,要求 P 至少为\_\_\_\_\_。





9. 解:



则: a1=0

a2=10

a3=110

a4=1110

a5=1111

信息熵  $H = -\sum P_i \log_2(p_i)$

$$= -(1/2 \log_2 1/2 + 1/4 \log_2 1/4 + 1/8 \log_2 1/8 + 1/16 \log_2 1/16 + 1/16 \log_2 1/16)$$

$$= 1/2 + 1/2 + 3/8 + 1/4 + 1/4$$

$$= 1 + 7/8 = 1.875 \text{ bit/字符}$$

a1—a5 码长分别为 1, 2, 3, 4, 4

则平均码长  $N = \sum P_i L_i = 1/2 \times 1 + 1/4 \times 2 + 1/8 \times 3 + 1/16 \times 4 + 1/16 \times 4 = 1.875 \text{ bit/字符}$

10. (略)

## 第5章 多媒体计算机硬件及软件系统结构

### 本章要点

- 了解具有代表性的多媒体计算机系统结构（硬件、软件），如 Intel/IBM 公司研制的 DVI 数字视频交互式多媒体计算机系统成功和失败的经验教训，对于理想的 DVI 系统怎样设计实现。
- 将多媒体功能集成到 CPU 芯片中，一类是以多媒体和通信功能为主，融合 CPU 芯片原有的计算机功能；另一类是以通用 CPU 计算机功能为主，融合多媒体和通信功能。把多媒体和通信功能集成到 CPU 芯片中的一些设计原则：在设计时采用国际标准；多媒体和通信功能的单独解决变为集中解决；体系结构的设计和算法相结合等。

多媒体计算机需要计算机交互式地综合处理声、文、图信息，尤其是图像和声音信息数据量大，处理速度要求高，用过去的通用计算机很难完成上述任务。例如，多媒体计算机中用得最多的静态图像压缩和解压缩算法，如果选用 Motorola 公司生产的 25MHz 的 68030 处理器芯片，完成一幅分辨率为  $512 \times 512 \times 8 \times 3$  的彩色静态图像的 JPEG 算法大约需要 15min，用 IBM PC/386 (33M) 完成上述算法大约需要 1min。为了较好地解决多媒体计算机综合处理声、文、图信息的问题，可以采用下述三种方法：

(1) 选用专用芯片设计专用适配卡单独解决，例如使用声卡解决声音的输入输出和实时编码、译码的处理问题；使用视频卡解决视频信号的输入、输出及多窗口的彩色键连问题；使用视频信号压缩编码和译码卡解决视频信号的压缩和解压缩问题；使用局域网的 ISDN 网适配卡解决局域网和远程网的多媒体通信问题。这种方法就是我们这一章中多媒体个人计算机 (MPC) 的标准和结构。

(2) 设计专用芯片和软件，组成多媒体计算机系统，综合解决声、文、图问题。比较成功的系统是 Philips/Sony 公司研制的 CD-I 系统和 Intel/IBM 公司生产的 DVI 系统。例如 DVI Action Media 750 II 系统，Intel 公司自己设计了 Intel 82750 PB 像素处理器及 Intel 82750 DB 显示处理器，还设计了三个专用的门阵电路 Intel 82750 LV 视频获取、SCSI 及 VRAM 控制接口门阵，82750 LH 主机接口门阵以及 82750 LA 彩色键连和声音接口门阵，组成了视频音频引擎 (AVE)。同时 Intel/IBM 公司还设计了 AVSS 和 AVK (音频视频子系统和音频视频核) 软件。

(3) 最后一种解决方案是把多媒体技术做到 CPU 芯片中。从目前的发展趋势看可以把它们分成两类：一类是以多媒体和通信功能为主，融合 CPU 芯片原有的计算功能，它们的设计目标是用在多媒体专用设备、家电及宽带通信设备上，可以取代这些设备中的 CPU 及大量的 ASIC 芯片，它们的代表产品是 Philips 公司设计制造的 Trimedia。另一类是以通用 CPU 计算机功能为主，融合多媒体和通信功能，它们的设计目标与现有计算机系列兼容，具有多媒体和通信功能，它们的代表产品是 Motorola 公司的 VE Comp 701 以及 Intel 公司的 MMX

技术。

### 5.1 光盘交互式多媒体计算机系统——CD-I

CD-I (Compact Disc-Interactive) 是由 Philips 和 Sony 公司设计制造的光盘交互式多媒体计算机系统。它是家电商品制造厂家瞄准家庭用户而设计制造的电视计算机 (Teleputer) 型多媒体计算机系统。它以价格低、存储量大 (650MB/片盘) 的 CD-ROM 作为数字信息的存储媒体, 配上 Motorola 公司生产的 68 系列 CPU 微机系统, 实现交互性, 组成了 CD-I 系统。

#### 5.1.1 CD-I 系统的硬件结构

CD-I 系统和 DVI (Digital Video Interactive) 系统的发展是源于光盘的发展 (详见图 5-1)。Philips 公司的研究人员将激光聚焦为微米光束, 成功地研制了能够记录和重放信息的光盘。1972 年 9 月 5 日他们向国际新闻界展示能够长时间播放电视节目的 Laser Vision 光盘系统。1978 模拟电视光盘的播放机正式投放市场, 光盘的直径为 30cm, 一片双面盘的播放时间为 2 小时。1982 年 3 月他们又成功地开发了数字光盘音响系统 CD-DA (Compact Disk-Digital Audio), 当时使用光盘直径为 12cm, 每片盘能播放 72min 高质量的音乐节目。1985 年 Philips 公司和 Sony 公司共同拟定了主要用于 PC 机外部存储 CD-ROM 光盘标准。为了能在计算机之间交换信息, 又制定了 CD-ROM 文件标准, 1988 年正式定为国际标准 ISO 9660。CD-ROM 投放市场后, Philips 公司和 Sony 公司于 1986 年, Intel 和 IBM 公司于 1987 年在光盘国际会议首先宣布和演示了 CD-I 系统和 DVI 系统。

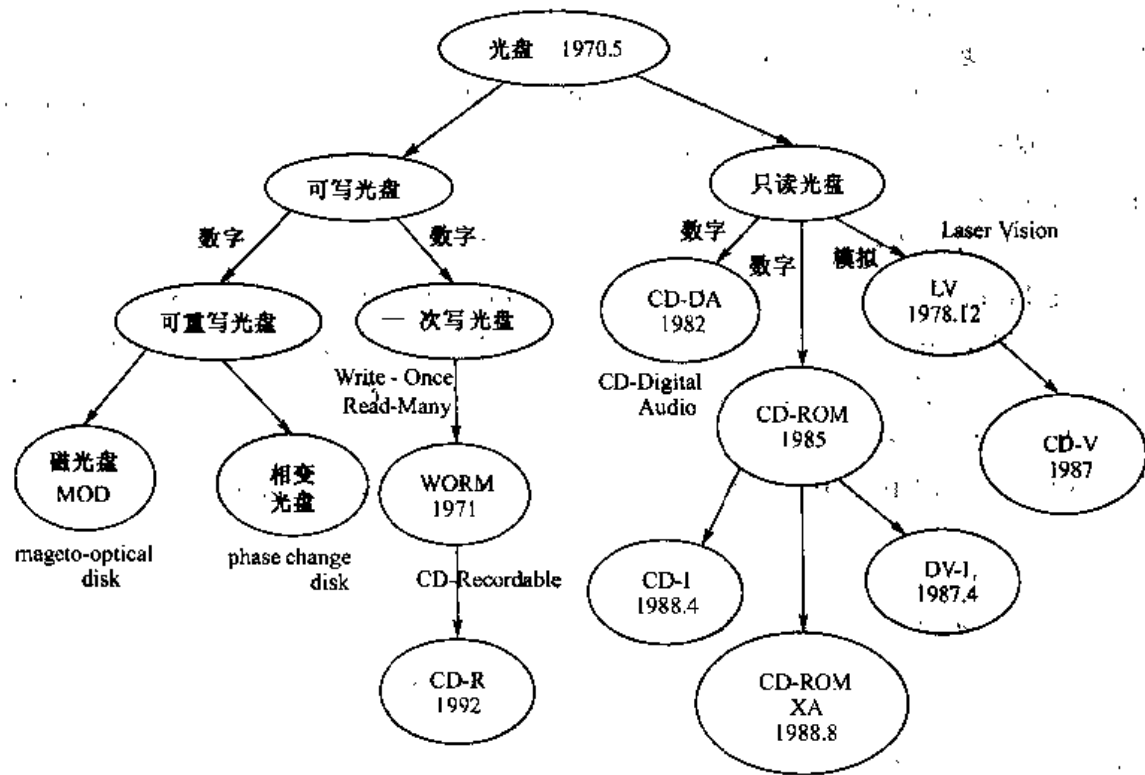


图 5-1 光盘发展系列

## 1. CD-I 光盘的数据格式

从系统结构出发,扼要回顾一下 CD-I 光盘的数据格式,对于理解 CD-I 系统的软硬件结构及系统设计原理是非常必要的。CD-I 光盘的数据格式是从 CD-DA 和 CD-ROM 光盘格式演变而来的。它和 CD-DA 光盘、CD-ROM 光盘都有三个区:导入区 (Lead-in Area)、节目区 (Program Area) 和导出区 (Lead-out Area),如图 5-2 所示。CD 盘上的信息均采用 EFM (Eight to Fourteen Modulation) 记录方式进行记录。由于光盘原始误码率较高,所以都采用能纠突发错误的 CIRC (Cross-Interleaved Reed-Solomon Code) 码。

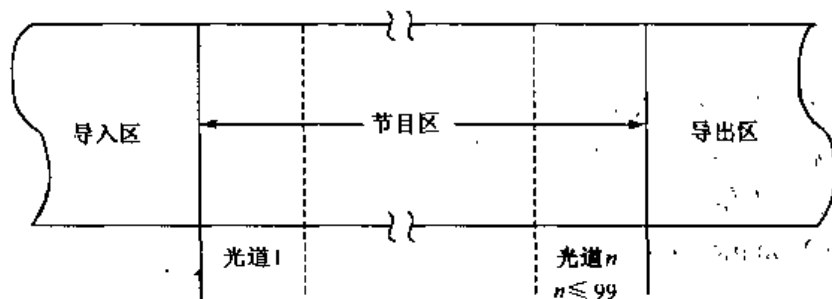


图 5-2 CD 盘结构

CD-I 光盘的导入区是由若干个空扇区组成的,这样做的目的是使识别节目区变得容易。

CD-I 光盘的节目区如图 5-3 和图 5-4 所示。图 5-3 表示只有 CD-I 盘结构,而图 5-4 表示 CD-I 光盘可以含有 CD-DA 光道的 CD-I 盘结构,这是为了和早期出现的 CD-DA 光盘格式兼容而设置的。CD-I 光盘和 CD-DA 光盘一样,可以有多到 99 条光道,从 1~99。对于包含有 CD-DA 光道的 CD-I 光盘,其第一条光道必须是 CD-I 光道,而且任何一条 CD-I 光盘上的 CD-DA 光道最多不超过 98 条。而一条光道的长度可以是 300 个扇区 (播放时间 4s) 和 325000 个扇区 (相当于最长的超级 HiFi 播放时间 72min) 之间的数。

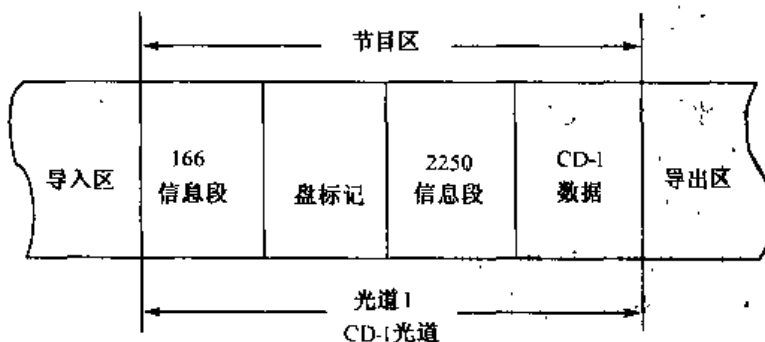


图 5-3 仅有 CD-I 光道的 CD-I 盘结构

CD-I 光盘的导出区或者是空扇区 (最后一条光道是 CD-I 光道时) 或者是无声的帧 (Frame) (最后一条光道是 CD-DA 光道)。

现在再回到节目区。节目的开头,也就是光道 1 的开始部分有 166 个信息扇区 (Message Sector),它仅含有 CD-DA 信息。2250 个信息扇区 (相当于播放时间 30 秒) 记录有敬告用户的一段话。当 CD-I 光盘在早期的 CD-DA 播放机上开始播放时能听到敬告用户的话,而在 CD-I 系统中则跳过这一段。

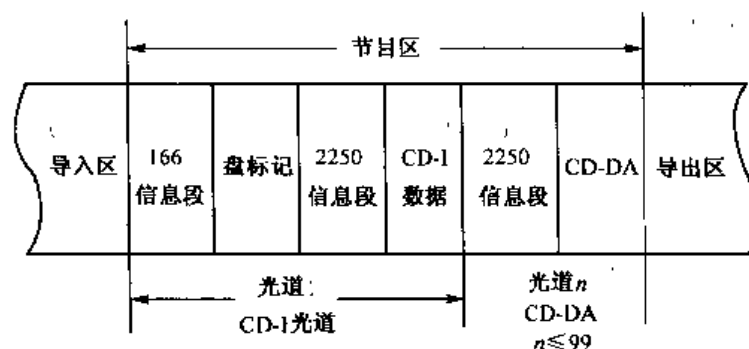


图 5-4 有 CD-I 和 CD-DA 光道的 I 盘结构

当 CD-I 光盘放到 CD-I 系统中播放时，系统首先读的扇区是光盘的标号（Disc Label）扇区。这个扇区包含 CD-I 盘上所有文件描述符、内容、大小、创作者等，必须加载到系统中软件模块的位置以及存取这些文件的路径表。光盘标号有三个记录组成：文件结构卷描述符（File Structure Volume Descriptor）、

引导记录（Boot Record）和终结记录（Terminator Record）。

路径表示于表 5-1，而目录结构示于图 5-5。每张光盘必须要有一张路径表，以提供盘上所存的目录结构索引，它跟着光盘标号，其位置在光盘标号中给出。每一个表目都包含如下三个信息组：

- 目录文件的位置。
- 父目录名。
- 目录名。

表 5-1 路径表

相 对 位 置	父 目 录 号	目 录 文 件 名
1	1	Root
2	1	CMDS
3	1	Games
4	1	Library
5	3	Chess
6	3	Checkers
7	3	Globono
8	4	Frank
9	4	Gibson
10	9	Audio
11	9	Text
12	9	Video

每目录都是一个文件，它包含文件描述符记录。

CD-I 盘上的所有数据都以文件的形式存放，任何一个文件都可以通过盘上的路径表取出。每个文件都有文件描述符记录存放于文件目录中。文件描述符记录包含有文件名、文件号、文件大小、地址、拥有者、属性、交叉存取因子、读取权限。

文件分为目录文件、实时文件和标准文件。

CD 数据可以两种方式存放于盘上，这两种方式是 Mode 1 和 Mode 2。CD-I 在 Mode 2 中定义两种专门的数据格式：Form 1 和 Form 2，两者的差别详见下节。所有 CD-I 的数据都是用 Mode 2 记录。而光盘标号中的数据则是用 Mode 2 中的 Form 1 来记录。因为它有 EDC（Error Detection Code）和 ECC（Error Correction Code）码，CD-I 系统利用它们可以获得误码率小于  $10^{-13}$  的数据。

## 2. CD-I 扇区格式

CD-I 数据分成离散的数据单元，称为扇区，其结构类似于 CD-ROM 的 Mode 2，但和它不一致。这些扇区包含有音频、视频、计算机数据以及需要系统实时处理的重要信息。一个 CD-I 扇区相当 CD-DA 中的一帧，系统以每秒 75 扇区的速度读 CD-I 盘。

每个扇区有 2352 个字节，由三个基本部分组成：Header、Subheader 和 Data Area，如

图 5-6 所示。Header 有 12 个同步 (SYNC) 字节；以时、分、秒作地址的 3 个地址字节以及一个 Mode 字节，其值总是为 2。

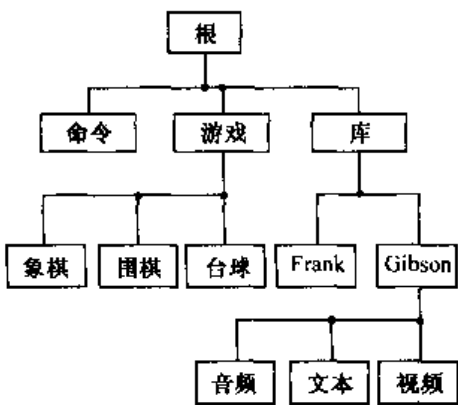


图 5-5 目录结构

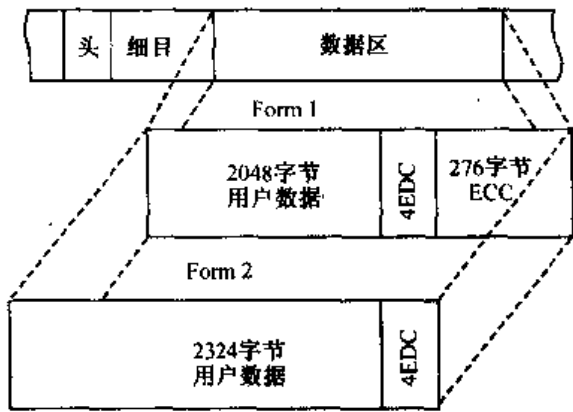


图 5-6 CD-I 扇区格式

Subheader 有 8 个字节，它进一步说明该扇区的用户数据：

- 数据类型。音频、视频或计算机数据等。
- 格式的形式。Form 1 或 Form 2。
- 触发位。记录开始、文件结束、实时性等。
- 数据的编码信息。如 ADPCM (Adaptive Differential Pulse Code Modulation)，CLUT (Color Look Up Table)，DYUV 等。

数据区的结构有两种：Form 1 和 Form 2。Form 1 不仅有 EDC 码而且还有 ECC 码，用来存放应用程序、控制数据、文字等要求不允许出差错的数据。Form 2 只有 EDC 字节用于存放误码率要求不太苛刻的数据，如音频和视频数据。利用 Form 2 来存放这类数据，其目的是充分利用光盘的存储空间。

### 3. CD-I 音频子系统

CD-I 的基本系统有四种标准音质的运行方式和一种非实时的语音音质运行方式。CD-I 除继承 CD-DA 超级高保真音质运行方式外，还有 A、B、C 三个音质等级的运行方式。A 级相当于 Laser Vision 的音质，即相当于高保真音质，B 级相当于 FM 调频广播的音质，C 级相当于 AM 调幅广播的音质。这四种音质的语音都是实时的语音。

非实时语音音质是文本到语音编码转换而成的音质。CD-I 有两种接口用来辅助编码这种音频信息，它们是上层接口和下层接口。上层接口是处理器默认的字符集，下层接口是对 8 位 PCM 数据进行实时译码。这两种接口之间的转换由微处理器控制。

CD-DA 超级高保真声音信号采用脉冲编码调制 PCM (Pulse Code Modulation)，用 44.1kHz 的采样频率对信号进行采样和编码。而 CD-I 的声音则采用自适应差分脉冲编码调制 ADPCM 技术进行压缩编码，把声音信号量化成 8 位或 4 位的二进制数，这就节省了存储空间，在声音的质量和表示声音的信息量之间进行折衷。ADPCM 编码是对被采样信号的预测采样值和实际的采样值之差进行量化，并把自适应预测和自适应量化结合起来使用，从而达到压缩音频数据的目的。

对 CD-I 中的信道有两种用法：一种是用于延长播放时间，如图 5-7 所示。另一种是用于实时获取并行音频数据流，如图 5-8 所示。

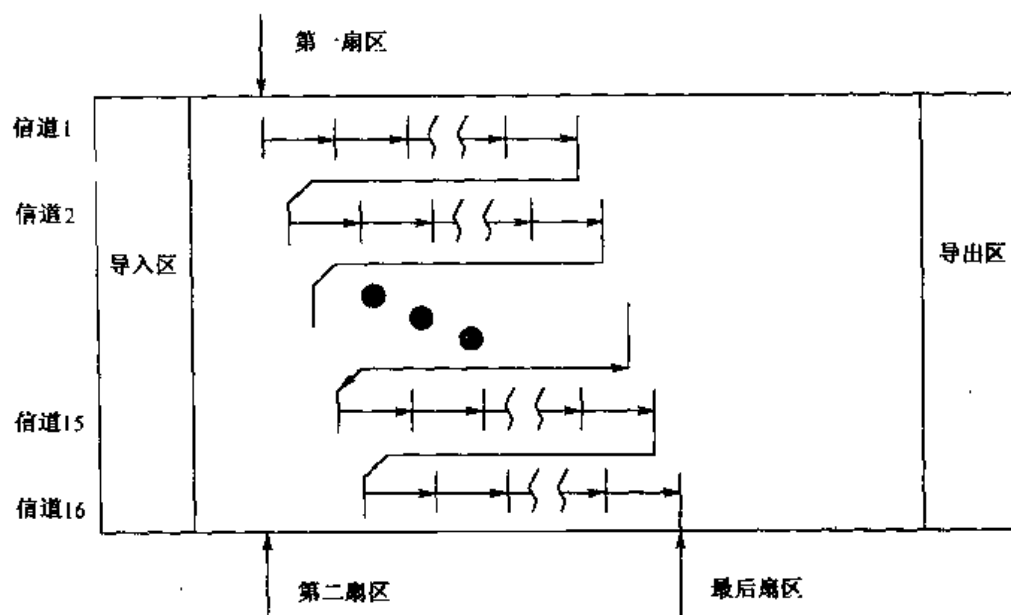


图 5-7 信道用于延长播放时间

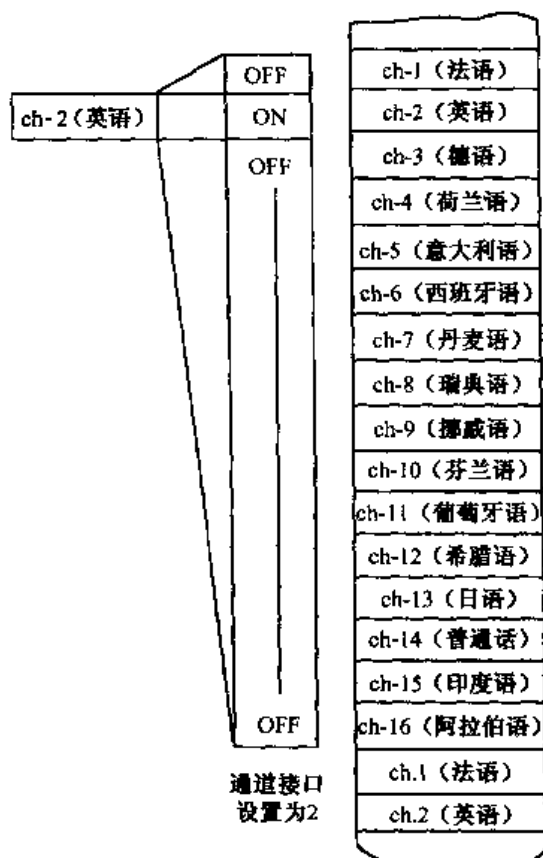


图 5-8 多信道实时获取数据

CD-I 的音频扇区结构如图 5-9 所示。音频数据保存在 Form 2 的实时扇区中。Subheader 中的 8 个字节含有加重语气设置信息、每个声音样本的位数 (A 级的位数为 8, B、C 级为 4)、采样速度 (A、B 级为 37.8kHz, C 级为 18.9kHz) 以及编码信息, 它表示该扇区所存

的信息是单声道还是立体声。

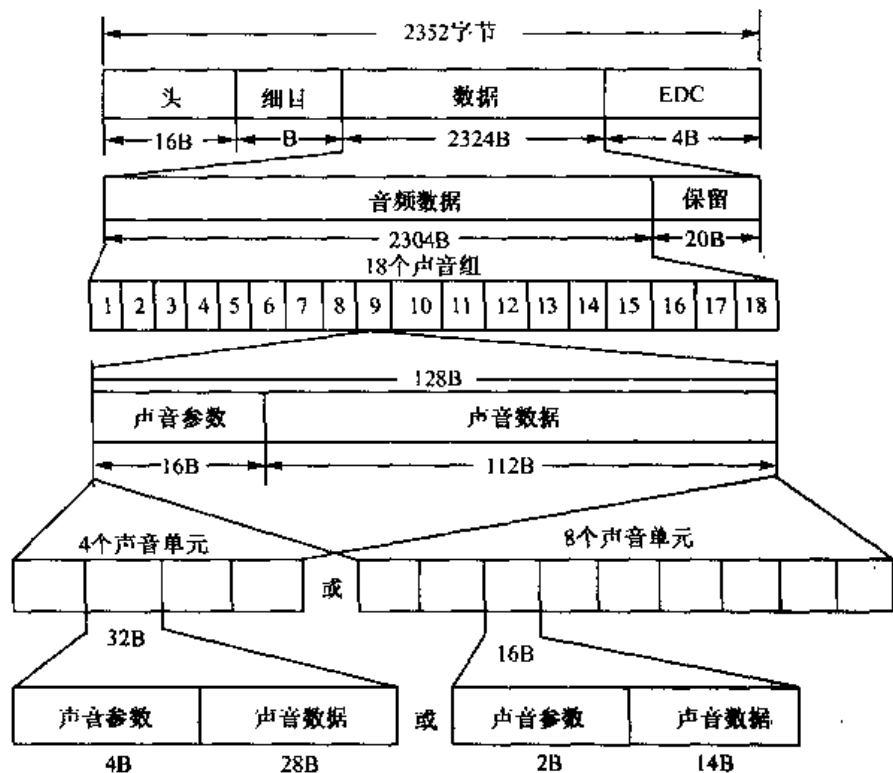


图 5-9 音频扇区结构

每个扇区由 2304 个字节的数据和 20 个空余字节组成。2304 个字节分成 18 个声音组，每个组由 128 个字节组成。128 个字节中的 16 个字节用来存放声音参数，其余的 112 个字节存放实际的声音数据。声音参数包含滤波和音域值，它们是在编码过程中获得的一个声音组的最佳参数。

声音数据译码和控制是由 CD-I 音频处理器子系统来完成的。一种音频处理器子系统示于图 5-10。它由 ADPCM 译码器、声音处理单元、特技处理器、控制器和声音存储器组成。

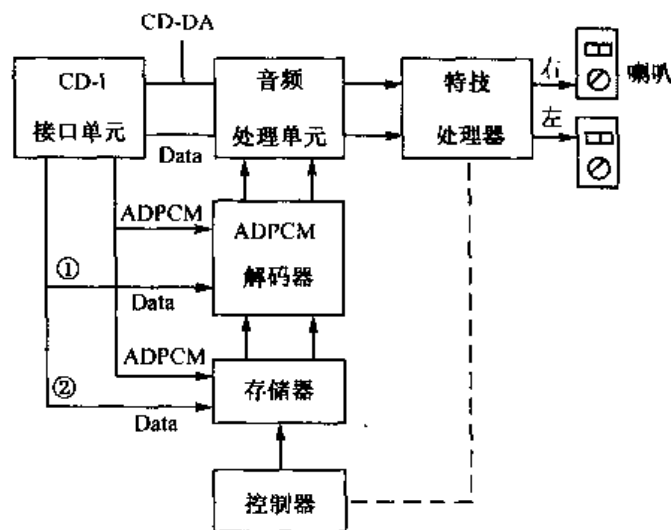


图 5-10 音频处理器子系统



ADPCM 数据可以从 CD-I 接口单元直接送到 ADPCM 译码器（图中的①），或者间接从声音存储器送到 ADPCM 译码器（图中的②）。ADPCM 译码器对 A、B、C 级的声音数据进行译码。ADPCM 译码器输出 PCM 声音数据到音频处理单元。而来自光盘的 CD-DA 资料是直接进入音频处理单元的。

音频处理单元的作用像一个简单的两信道声音混合器，它的两个输入信道——立体声信道或者单声道，能把输入信道的信号混合后送到左右两个输出信道。它实质上是一个数模转换器，但当它和特技处理器的功能结合起来时，在 CPU 的控制下就像一个音量可控的声音混合器，特殊的音响效果可以利用软件来产生而不必预先记录到盘上，这就节省了存储空间。

4. CD-I 视频子系统

CD-I 系统提供了三种不同的图像分辨率。

正常分辨率的图像质量相当于广播电视接收机得到最好的图像质量。双倍分辨率的图像质量相当于在计算机监视器上可获得的最好的图像质量。高分辨率的图像质量相当于演播室中产生的最好的图像质量。

由于显示一幅图像需要很大的数据量，例如显示一幅分辨率为  $360 \times 240$  像素的彩色图像，每个像素用 16 位的二进制数表示，就要  $360 \times 240 \times 16/8 \approx 172800$  个字节，而用 CD-I 中的 Form 2 扇区格式，每秒钟的数据传输率为  $2324 \text{ 字节/扇区} \times 75 \text{ 扇区/秒} = 174300 \text{ 字节/秒}$ ，这就需要约 1s 的时间才能从光盘中读出一幅正常分辨率彩色图像的数据。为提高图像的显示速度和减少图像的数据量以节省存储空间，针对自然图像、图形、动画等各有特点的图像，CD-I 采用了四种基本的压缩技术，说明如下：

(1) 一维的 DYUV 编码。主要用于自然图像，如彩色照片等。这种编码方法是把亮度信号 Y 和两个色差信号 U、V 的变化值，即相邻两个像素分量值之差进行量化编码，逐行转换成二进制数，译码后每个分量都变成 8 位二进制数，故有  $2^{24} \approx 16\,772\,216$  种不同的颜色。DYUV 图像的数据格式如图 5-11 所示。

DU	DY <sub>1</sub>	DY	DY <sub>2</sub>
4位	4位	4位	4位

图 5-11 DYUV 图像编码

(2) RGB 5:5:5 编码。主要用于高质量的图形。对组成彩色图像的三种基色：红（Red）、绿（Green）、蓝（Blue）分量都用 5 位二进制数表示，每个像素用 16 位二进制数表示，如图 5-12 所示。这样表示的图的颜色数共有  $2^{15} \approx 32\,768$  种不同的颜色。

T	Red	Green	Blue
1位	5位	高2位   低3	5位

1: Transparency

图 5-12 RGB5:5:5 图像编码

(3) CLUT (Color Look-Up Table) 编码。主要用于要求变化较快的图形，便于对图像进行管理。CLUT 的内容放在光盘上。CD-I 系统中含有几种不同的 CLUT 方式如图 5-13 所示。CLUT8 使用 8 位地址去查找 256 种颜色的调色板寄存器，CLUT7 有 128 种颜色，而 CLUT4 有 16 种颜色。

(4) 一维行程码。主要用于动画这类图像。动画上面有许多颜色相同的块，而且画面上的颜色数目及颜色的变化数目也有限，因此用行程编码压缩图像数据很有效。行程码用 CLUT 去定义颜色，然后定义在像素颜色改变之前具有相同颜色的像素数目。CD-I 定义了两种 RL (Run-length) 编码：RL7 和 RL3，分别示于图 5-14。RL7 在正常分辨率的图像中，使用 1 个字节表示颜色，另一个字节表示行程长度。

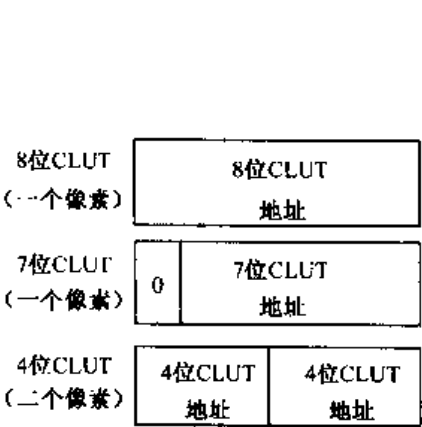


图 5-13 CLUT 图像编码

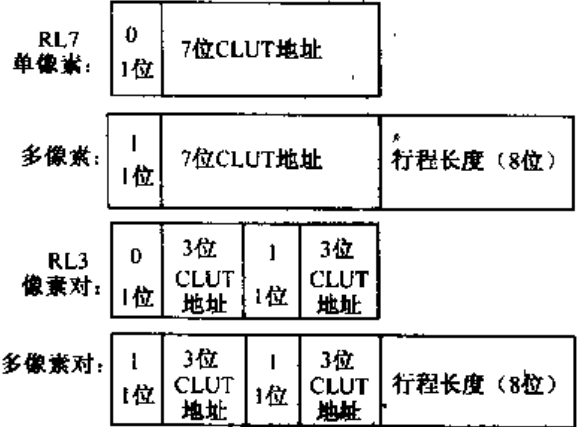


图 5-14 RL 图像编码

用户看到的 CD-I 图像是由多个图像平面合成的，就像 CD-I 中的数字音响一样。CD-I 音频子系统中有左右 ( $L_{in}$  和  $R_{in}$ ) 两个输入信道，可以想像成两个平面，在 CPU 控制下，根据需要可把音频数字信号混合后送到左右 ( $L_{in}$  和  $R_{in}$ ) 两个输出信道，经喇叭转换成优美动听的声音。而 CD-I 的图像是由四个图像平面组成的，如图 5-15 所示。平面 1 是一个  $16 \times 16$  像素的彩色光标平面，平面 2 和平面 3 是全屏幕图像平面，平面 4 是背景平面。在软件的控制下，这些平面上的图可以按各种要求叠加生成一幅画面显示在荧光屏上。

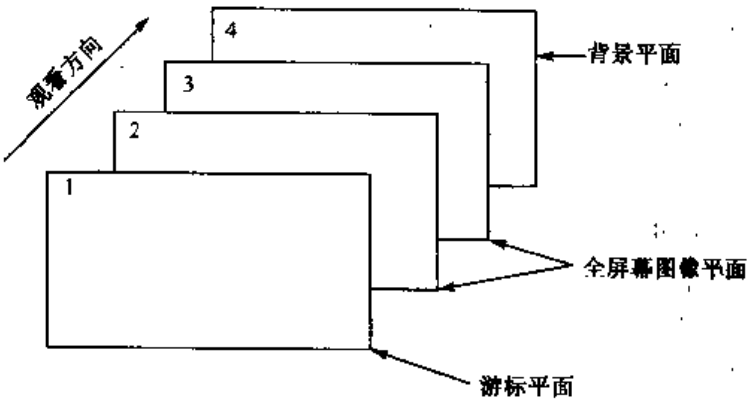


图 5-15 CD-I 的图像平面

经过编码显示在 CD-I 光盘上的数字视频信号由 CD-I 视频子系统来还原。该子系统如

图 5-16 所示。它有两个由 RAM 构成的存储体 0 和存储体 1 与存取控制器相连。存储体 0 存储图像平面 A（即平面 2）的信息，存储体 1 存储图像平面 B（即平面 3）的信息。这两个存储体也作为声音数据存储器和系统存储器来使用。而子系统内的存取控制器是用来控制存储体、两个译码器与主系统总线之间的信息交换。该子系统具有如下几个基本功能。

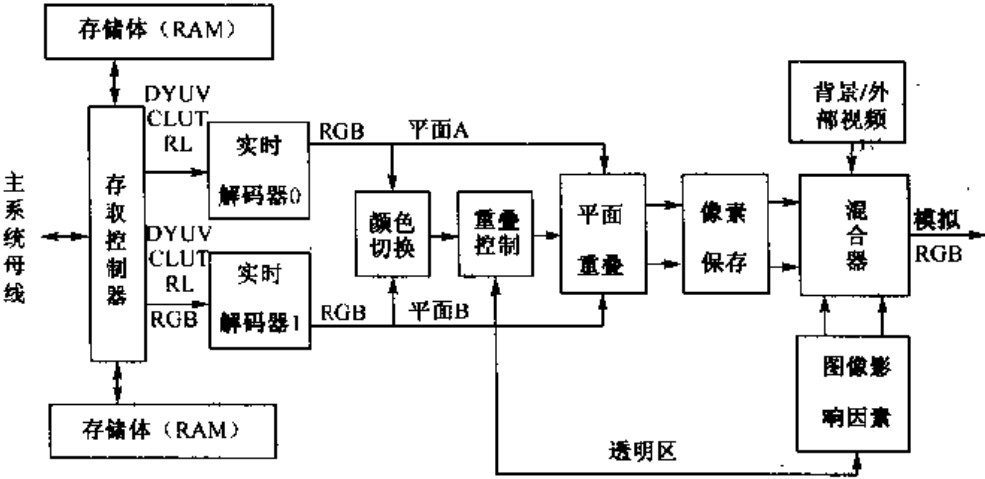


图 5-16 CD-I 视频处理器子系统

● 译码：

DYUN→YUV→RGB 8: 8: 8

CLUT→RGB 8: 8: 8

RL CLUT→RGB 8: 8: 8

RGB 5: 5: 5→RGB 8: 8: 8

- 像素的颜色切换，用于在图像平面上产生透明区。
- 重叠图像平面。
- 像素保存，用于缩放图像，改变图像分辨率而不改变图像大小那样的马赛克效应。
- 混合图像，用于组合每个图像平面上的像素值变成一幅图。
- 把数字图像信号转换成模拟 RGB 信号。

如图 5-16 所示，实时译码器 0 用于译 DYUV、CLUT 和 RL 编码图像，而实时译码器 1 用于译 DYUV、CLUT、RL 和 RGB 编码图像。每个译码器输出 R、G、B 三个彩色分量，每个分量是 8 位。对于 DYUV、CLUT 和 RL 码，每个译码器输出的 R、G、B 分量至少是 6 位。对 RGB 5:5:5，每个 8 位分量的最低 3 位是 0，高 5 位是分量的像素值。

图像平面 A 和 B 的重叠可以是 A 在 B 后，或者相反，用图中所示的颜色切换和在图像平面上产生透明区来实现重叠。平面 A、B 上的图像在混合器中混合，而且每个平面的亮度也可以由图像影响因素来控制，这就允许做数字特技，如淡入淡出。

光标平面上有 16×16 个像素，每个像素都可以设置为透明的或彩色的，光标平面和背景平面在混合器与平面 A、B 相混合后转换成模拟的 RGB 信号。

对于 DYUV 编码图像，实时译码器的 DYUV 部分示于图 5-17。它的编码以两个像素作为一个基本单元，其长度为两个字节；每对像素有两个 ΔY 值，一个 ΔU 和一个 ΔV 值，在译码器中把它们转换成非线性量化值，连续的 Δ 值相加得到 Y、U、V 分量的绝对值。对每

一行，如果该行不是黑色的，则必须提供起始点的  $Y$ 、 $U$ 、 $V$  绝对值。这些绝对值已单独编码并记录在光盘上。 $U$ 、 $V$  值分别插到中间的像素点上。最后将  $Y$ 、 $U$ 、 $V$  作矩阵变换产生模拟的  $RGB$  信号供显示用。

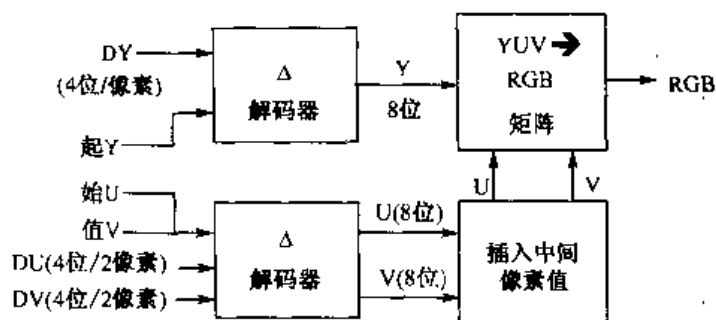


图 5-17 实时解码器的 DYUV 部分

对于  $RGB\ 5:5:5$  编码图像，译码时使用两个存储体，产生单幅图像。

对  $CLUT$  编码图像，译码时使用两个存储体，产生单幅图像。

对  $CLUT$  编码图像， $CLUT8$  和  $CLUT7$  是每个字节代表一个像素， $CLUT4$  和  $CLUT3$  每个字节代表 2 个双倍分辨率的像素。 $CLUT7$  的图像可以在两个平面上， $CLUT8$  的图像只能在平面 A 上。

对于  $RL$  编码，需指出的是，所有行末端的  $RL$  码在编号时游程长度都设置为 0，这就意味着在译码时颜色必须延续到该行的终点。

由于译码器只有一张 256 个单元的颜色查找表，所以有些图像不能同时在平面 A、B 上产生，这是受速度的限制所致。平面 A、B 可得到的图像与编码之间的关系如图 5-18 所示。

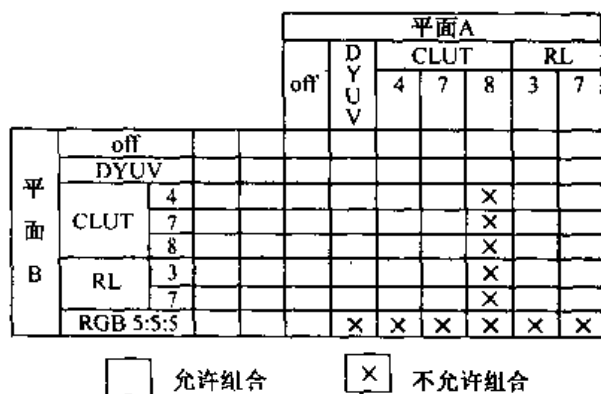


图 5-18 平面 A、B 上编码组合

加载图像数据、 $CLUT$  的值、定义的图像透明区、图像影响因素以及显示图像等任务，都是由  $CD-I$  中定义的显示控制程序（DCP）来完成的。DCP 由一系列显示控制指令组成，它们分布在整个  $CD-I$  光盘上或者在一个区域上，控制行扫描及场扫描。这些指令放在两张表里，一张叫场控制表（FCT），每一场执行一次。另一张叫行控制表（LCT），每条显示行都有单独的指令。

图像平面 A、B 都有场控制表和行控制表。场控制表可容纳多达 1024 条指令，用来建立每场开头的显示参数以及把数值加载到颜色查找表中去。行控制表可以容纳多达 8 条指

令，从一行到另一行的扫描过程中，这些指令允许改变显示参数。

行控制表的最基本的用途之一是用来为一行图像的开头提供显示的起始地址，该地址又可以放到存储器中的任意位置。这一点对于屏幕上部分或整屏图像的滚动以及数字特技的实现显得特别重要。

显示控制程序的重要应用之一是产生图像的透明区，作为一个例子示于图 5-19。在这个例子中有三个重叠的透明区（分别为 Matte 1、Matte 2 和 Matte 3），和一个图像影响因素区 ICF。假如我们仅考察一条显示行，就会发现需要 8 条行控制命令才能实现这种显示效果：

- A：开始 Matte 1。
- B：改变 ICF 的值为新值。
- C：开始 Matte 2。
- D：结束 Matte 1。
- E：开始 Matte 3。
- F：结束 Matte 2。
- G：改变 ICF 的值为原值。
- H：结束 Matte 3。

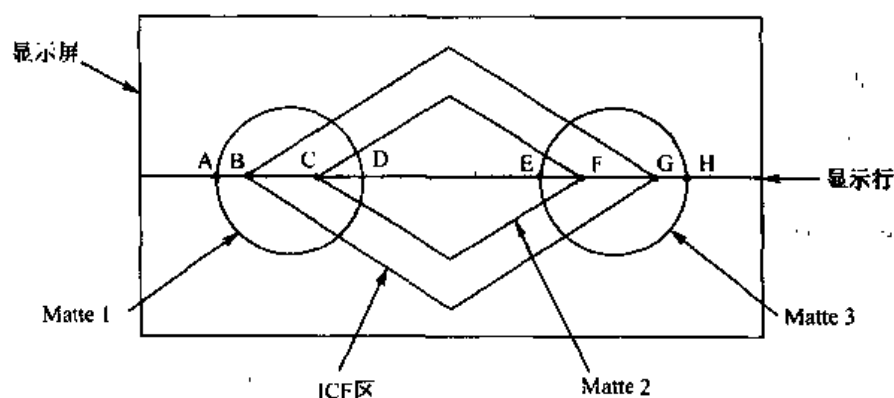


图 5-19 显示控制程序应用实例

此外，DCP 还提供许多其他的控制指令，如运动目标与显示扫描的同步等。DCP 是光盘操作系统中的一个程序，这里介绍的目的是想使读者对视频子系统的工作原理有更深入的理解。

## 5. CD-I 基本系统

在介绍 CD-I 基本系统的结构之前，需说明一点的是，在定义 CD-I 基本系统时充分考虑了允许所有 CD-DA 盘都能在 CD-I 系统中播放。这一点对理解 CD-I 系统是很有帮助的。

CD-I 基本系统的结构如图 5-20 所示。它由以下几部分组成：

- 光盘播放机。
- CD-DA 控制器/接口单元。
- CD-I 接口。
- CD-I 音频处理子系统。
- CD-I 视频处理子系统。

- MPU (MicroProcessing Unit)。
- DMA 控制器。
- RAM。
- 存取控制器。
- NVRAM。
- CD-RTOS、CSD (见光盘实时操作系统一节) 和字体模块 ROM。
- 定位设备。
- 时钟/日历。
- 键盘 (选件)。

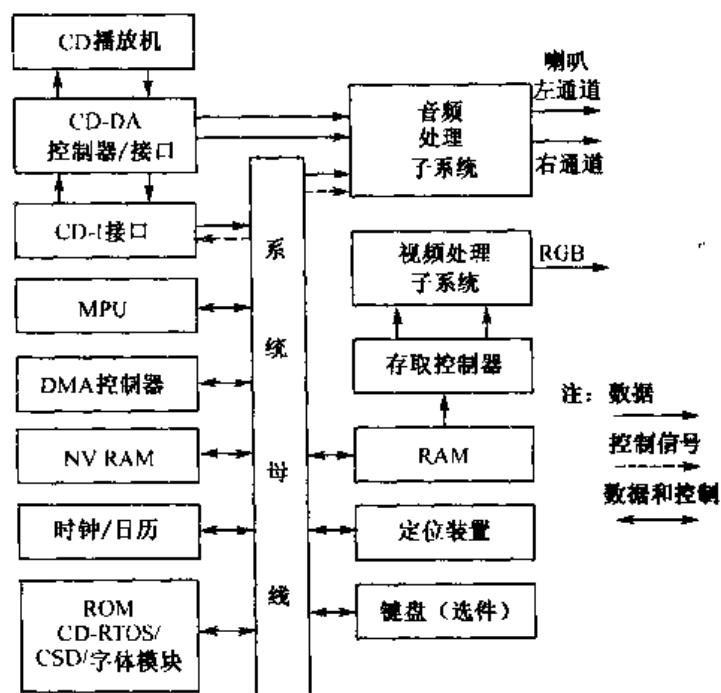


图 5-20 CD-I 基本系统

光盘播放机可以在 3 秒钟之内读取 CD-I 或 CD-DA 光盘上任何部分的数据。播放机在程控下可以暂停、连续、停止、弹出光盘。

从光盘读出的数据经过 CIRC 错误校正之后，由 CD-DA 控制器/接口单元把数据有选择地送到音频处理子系统（如果数据是 16 位 PCM CD-DA 数据）或 CD-I 接口（如果数据是 CD-ROM 或 CD-I 数据）。它也控制光盘播放机的运行。

CD-I 接口把来自 MPU 的控制信息送到 CD-DA 控制器/接口单元，也把经过筛选的数据送到系统总线。于是 CD-I 接口要分解来自光盘的交叉数据，产生中断信号并把整理的数据送到相应的处理单元。所有这些事情都是通过实时解释扇区细目 (Subheader) 中的信息来控制的。

CD-I 音频和视频处理子系统已在前面作了详细介绍。有一点要补充的是，视频子系统不能处理非隔行扫描的高分辨率图像。

CD-I 基本系统能够读和译 CD-DA PCM 数据，这种数据可以来自 CD-I 或 CD-ROM 光盘上的 CD-DA 光道，或者来自 CD-DA 光盘上的 CD-DA 光道。

CD-I 光盘上的程序代码必须是 M68000 可执行目标码，总的 RAM 至少为 1MB，并要组织为至少两个存储体，每个为 512KB。而 NVRAM 的容量至少要 8KB。

CD-I 基本系统至少要有有一个 x-y 定位设备，它能够指定正常分辨率图像的每个像素。此外，它至少要有两个触发按钮。

CD-I 基本系统的时钟/日历可用电池作后备电源，至少要以秒做单位，除年、月、日、时、分、秒外，还能计算出闰年。

所有 CD-I 基本系统必须有默认的字符集，并且至少要有 4 种字体放在 ROM 中。ROM 中还装有 CD-RTOS 和库模块。

所有 CD-I 基本系统中的设备都是中断驱动的。

## 5.1.2 CD-I 光盘实时操作系统

计算机系统的核心是操作系统，它控制整个系统的硬件和其他软件的协调运行。CD-I 有它自己的光盘实时操作系统 CD-RTOS (Compact Disc Real Time Operating System)，它源于高性能的 OS-9 实时操作系统，是用 68000 汇编语言写成的。

### 1. CD-RTOS 的特性

CD-RTOS 是多任务实时操作系统。CD-I 系统必须支撑多种数据类型和数据流，即音频、视频、文字、控制 and 应用程序，也还要支撑多种 I/O 设备，如键盘、像素定位装置、显示器等。这些数据流也许要调用几个不同的进程，这些进程也许是“同步”的，也许是“异步”的，而且要在限定时间内完成。CD-RTOS 为此提供了多重处理的能力。

CD-RTOS 的结构是模块化结构。程序由机器指令和数据组成，指令代码以模块的形式保存，一个程序由一个或多个模块组成，几个程序也可共享某些模块，这些模块有些存放在光盘上，有些存放在 ROM 中，有些在运行时调到 RAM 里。每个模块都有自己的名字，操作系统根据模块的名字来查找。这些模块可以被调到存储器，也可以从存储器删除以便节省存储空间。CD-RTOS 由许多模块构成，这就具有很大的灵活性，当系统硬件有变化时，只要改变或增加几个模块就可以，无需对操作系统做大的更改。

CD-RTOS 有设备独立的 I/O 接口。每个文件和设备都要命名以便打开它们。但打开操作返回的是路径号，所有程序对文件的后续操作都用这种路径号，程序能从中得到详细的 I/O 信道的信息，但许多场合下甚至无需知道究竟是和哪个设备交换信息。

CD-RTOS 能够处理多级树形结构的盘目录。一张盘在格式化时都产生一个根目录。根目录和其他子目录下的文件用路径号来命名。CD-RTOS 支撑两种默认的目录，一个是为执行文件而建立的目录，另一个是为其他所有文件而建立的目录。

CD-RTOS 是中断驱动的系统。

### 2. CD-RTOS 的结构

CD-RTOS 是处于应用程序和硬设备之间的程序，它的结构如图 5-21 所示，它主要由四部分组成。

(1) 系统相关库。库中含有用户所必需的库程序，如高级存取和数据同步程序，以及数学函数、I/O 和其他程序，这些程序最重要的是同步程序。

(2) CD-RTOS 核。它处理多任务，任务控制、进程通信、存储器管理、中断处理、系统服务请求、I/O 调用、任务同步等，是 CD-I 的心脏。

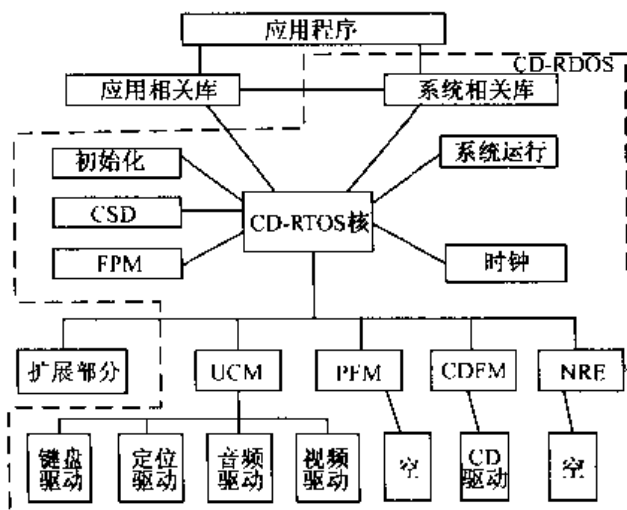


图 5-21 CD-RTOS 结构

(3) 管理程序。管理程序主要有：

- 光盘文件管理模块 CD-FM。
- 用户通信管理模块 UCM。
- 非易失 RAM 文件管理模块 NRF。
- 流式文件管理模块 PFM。

这些管理程序支持 CD-I 视频处理设备、定位设备、CD-I 音频处理设备以及盘的 I/O 和确定盘的最佳存取方案。这些程序的上层是和硬件无关的，而下层是和各种设备驱动程序相连接的。

(4) 设备驱动程序。包括键盘、定位设备、音频子系统、视频子系统、光盘播放机等驱动程序。

系统结构状态描述符 CSD (Configuration Status Descriptor) 是 CD-I 系统的一个组成部分，它允许应用程序去找出 CD-I 系统中有哪些设备可用，以及这些设备处在什么状态。系统中的每台设备在 CSD 中都有表目，CSD 存放在 RAM 或 ROM 中，而用户的设置放在 NVRAM 中，每条表目称为设备状态描述符 DSD (Device Status Descriptor)，它由 4 部分组成：

- 设备类型。
- 设备名称。
- 活动状态。
- 设备参数集。

CSD 中描述的设备包含了 CD-I 基本系统的资源，如视频译码器、音频译码器、显示类型、x-y 定位设备以及扩充的设备，如软盘驱动器、打印机、调制解调器等。

文件保护器 (File Protection Mechanism) 用来保护光盘上的文件不被随便读取。它允许光盘物主保护多达 32 个不同文件。这就意味着光盘包含多个应用程序时，可以通过单独的存取权限代码对它们进行分别控制；这也就意味着只有用户对每一个应用程序付了钱才能获得光盘上的全部信息，用户付一个应用程序的钱就只能用一个。CD-RTOS 核中的保密方法是把存取权限代码记录在光盘上，播放时从光盘译出的码和用户付钱后得到的存取权限代码



进行比较,以确定是否可使用该程序。

### 3. 文件管理程序

文件管理程序是介于 CD-RTOS 核和设备驱动程序之间的程序,用来快速处理输入输出。主要的文件管理程序有三个,下面加以说明。

(1) 光盘文件管理程序 CDFM (Compact Disc File Manager)。该程序把来自应用程序的高级命令转换成 CD 播放机设备驱动程序中的命令,它存取和解释光盘文件系统,实现必要的存取保护,并安排光盘的存取。CDFM 处理的服务请求有:

- 打开指定文件的路径。
- 改变目录。
- 改变当前文件或记录的指针。
- 读文件。
- 获取状态信息。
- 设置状态信息。
- 执行指定操作。
- 关闭文件。

CDFM 中有一条专门的命令用来播放 CD-DA 数字音响光盘。该命令直接把 CD-DA 数据引导到音频处理单元。

(2) 用户通信管理程序 UCM (User Communication Manager)。它管理和用户进行交互的四个主要设备:键盘、x-y 定位设备、音频和视频设备。

UCM 的视频管理模块支持两个图像平面,支持在存储器中产生和管理多幅图像和多种字体,可以同时激活 ROM 中的四种字体,提供基本的画图和文字处理功能。实现这些功能的基本思想是在存储器开辟一个区,称为画图区,管理程序对这个图像区进行各种操作。

UCM 的声音部分控制 ADPCM 声音的数据流和声音数据的输入输出,以及 MPU 到音频子系统的控制信号。这个程序也支撑非实时的声音输出和应用程序产生的声音,辅助管理存储器中的声音数据。实现这些基本功能的思想也是在存储器中开辟一个区,称为声音区,管理程序对这个声音区进行各种操作。

(3) 非易失性 RAM 文件管理程序 NRF (Nonvolatile RAM File Manager)。它为应用程序提供一个文件编排系统,把文件写到 NVRAM 中。它只提供一个基本的目录结构而无子目录。文件名由应用程序定义而无副本。由于 NVRAM 的容量所限(基本系统仅 8KB),所以应用程序使用它时要精打细算。

### 4. 同步和控制

从光盘上播放的声音是实时的,伴随有图像的声音或者伴随声音的图像也是实时的,而且是同步的。CD-I 系统中的同步,如声音和图像的同步可由好几种途径来获得:

- 软件定时程序引导产生定时信号。
- 用户输入产生的信号。
- 来自正在运行的另一个应用程序发生的信号。
- 在扇区 Subheader 中的触发位、记录结束位或者是文件结束位。
- 来自播放控制表 PCL (Play Control List) 缓存之一发出的缓存已填满信号。
- 请求记录传送已结束。

CD-RTOS 由播放控制结构来控制通过 CD-I 系统的数据；用扇区中的触发位设置信息和实时控制区 RTCA (Real Time Control Area) 的控制信息来实现同步。

在 CD-I 系统中，最小的不可中断的实时数据记录单元称为实时记录，实时记录组成实时文件。用来播放实时文件包含交叉存放的音频数据扇区、视频数据扇区和与程序有关的数据扇区。用来播放实时文件的播放系统调用数据文件号、信道号和数据类型（音频、视频或程序相关数据）来选择扇区，并且控制播放控制块 PCB (Play Control Block) 和播放控制表 PCL，并且，每当出现特殊事件时告诉应用程序。PCB 是一种数据结构，它为当前的或下一个实时记录的播放程序提供读盘信息，即输入信息流。而 PCL 也是一种数据结构，不过它是控制数据到达的目的地，即输出信息到相应的设备。

当数据读到系统中时，PCB 重新组织交叉存放的数据，并且把音频数据、视频数据和程序相关数据存放到相应的存储器中。PCB 指向音频 PCL、视频 PCL 和程序相关数据 PCL。

音频数据可以直接送到音频处理器或者送到声音存储区。PCB 提供信道选择屏蔽信号，用来表示哪个音频扇区数据要放到存储器，哪个音频扇区数据要直接输出到音频处理器。

相关数据 PCL 缓存包含有文字和控制数据。与实时记录有关的控制数据一般都放在实时记录开头的区，称为实时控制区 RTCA，如图 5-22 所示。RTCA 中的数据比实时数据先读入到系统中，并由实时记录解释程序 RTRI (Real Time Record Interpreter) 对它们作解释。RTRI 用来控制实时文件的再现和同步。RTCA 包含若干命令，用来控制实时记录的播放、加载音频和视频数据、管理和显示图像、控制声音的输出，以及实现各种实时数据类型的同步和与用户的对话。RTCA 中的命令能顺序执行或者以并行方式执行，或者以串并组合方式执行。RTRI 以普通文件的方式存放在 CD-I 光盘上，它是一个多任务解释器。

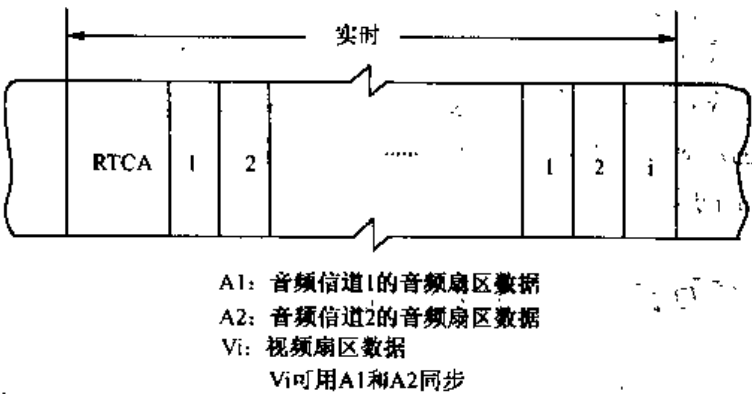


图 5-22 实时记录示例

CD-RTOS 支持的另一种同步方法是通过事件和事件处理来实现同步。事件有名称和数值，并且可以由任何一个进程存取。一个进程可以等待事件达到一个给定值或者给定范围的值，命令可以用来设置或增加事件的值。

5. 启动过程

当 CD-I 系统通电并把光盘插入以后，软件就执行启动程序。CD-I 系统的启动过程如图 5-23 所示。

## 6. CD-I 基本系统的主要技术指标

### (1) 播放时间:

理论值最长为 74min33s。

实际播放时间:

- 66min (CD-ROM 和 CD-I Form1)。
- 72min (CD-DA 和 CD-I Form2)。

### (2) 光盘数据率:

- 75 扇区/秒。
- 150KB/s (CD-ROM 和 CD-I Form1 用户数据)。
- 175KB/s (CD-I Form2 的用户数据)。

### (3) 光盘容量:

理论值最大为 335 475 扇区。

- 655MB (CD-ROM 和 CD-I 用户数据)。
- 745MB (CD-I Form2 用户数据) 实际容量为 324 000 扇区。
- 580MB (CD-ROM 和 CD-I Form1 用户数据)。
- 719MB (CD-I Form 2 用户数据)。

### (4) 图像:

活动图像尺寸 (525 行扫描系统, 正常分辨率, 最长播放时间 72min):

- DYUV 图像 132×88 像素, 占全屏的 13.5%。
- CLUT8 和 CLUT7 动画, 全屏。
- RL7、RL3 图像, 全屏。

全屏加载时间: 加载并显示 84.4KB 的全屏幕图像约需 0.495s。

以上简要介绍了家用交互多媒体系统 CD-I 的概貌。CD-I 是出现最早的交互式数字多媒体技术, 目标是瞄准家用市场。第一次发表是在 1986 年, 正式的商业产品直到 1991 秋才出现, 时经 5 年, 这也反映了 CD-I 系统软硬件的复杂性。CD-I 也将支持全屏幕全速的电视图像, 支持 MPEG 制定的动画图像和相关声音的编码标准。

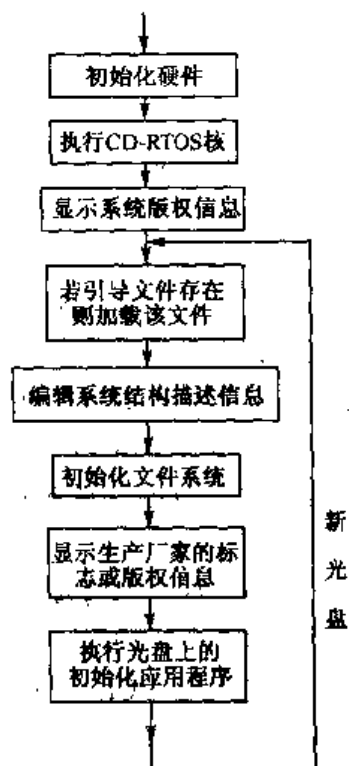


图 5-23 CD-I 系统的启动过程

## 5.2 数字视频交互式多媒体计算机系统——DVI

在 1987 年 3 月第二次 Microsoft CD-ROM 会议上首次公布了 DVI (Digital Video Interactive) 技术的研究成果, 1989 年 10 月 Intel 公司从 GE 公司买来了 DVI 技术, 1989 年 Intel 和 IBM 公司在国际市场上推出 DVI 技术的第一代产品 Action Media 750, 1991 年又在美国 Comdex 展示会上推出了 DVI 技术的第二代产品 Action Media 750 II, 它荣获了 Comdex 91 最佳多媒体产品奖 (Best Multimedia Product) 和最佳展示奖 (Best of Show)。Comdex 的最佳奖相当于计算机工业的奥斯卡 (Oscars) 奖。

### 5.2.1 DVI 系统中的视频音频引擎 (AVE)

II 型 DVI 系统 Action Media 750 II 是一个比较成熟的多媒体计算机系统, 如前所述, 它

获得了“Comdex 91”最佳多媒体产品奖和最佳展示奖。比起 I 型 DVI 系统，它有如下几点改进：

(1) 性能指针高。II 型 DVI 系统在硬件设计时采用了 82750 PB 像素处理器和 82750 DB 显示处理器专用芯片，82750 PB 和 82750 DB 比起 82750 PA 和 82750 DA 在运算速度上快一倍，而且在微码引擎和指令系统方面也有了明显的改进。

(2) 使用了专用的门阵电路。在 Intel 公司的 I 型 DVI 系统的基础上，将其周边逻辑设计成三个门阵电路：82750 LH 主机接口门阵；82750 LV VRAM/SCSI/Capture 接口门阵；82750 LA 音频子系统接口门阵。

(3) 将多块处理板变成一块处理板：最早的 DVI 系统由 5 块板组成，后来 I 型 DVI 系统由 3 块板组成，II 型 DVI 系统将它们集成在一块板上，视频和音频获取板也装在上面，只占一个 IBM PC 标准插槽。

II 型 DVI 系统的核心是视频算法和显示引擎，它们是 82750 PB 像素处理器，82750 DB 显示处理器和静止图像压缩编码和解压缩算法。同时还要完成把最后得到的位映像转换成在监视器能够显示的模拟 RGB 信号。

同视频引擎同时并行操作的是音频信号处理的硬件：音频信号处理器，D/A 转换器以及模拟滤波器。它们组成了音频信号处理子系统。它执行音频信号的压缩编码和解压缩、数字语音信号到模拟语音信号的转变等任务，最后在外部分放大器和音响设备播放。

为了支持这些子系统，大量基本的数据必须在 DVI 的 VRAM 和其他外设、主机之间传送，例如主机 CPU，获取子系统等和 VRAM 传送数据。在 DVI 系统中数据通信信道是具有多路开关的数据和地址总线，也称为“DVI 总线”。不仅主机能够使用这个 DVI 总线与每个 DVI 子系统（视频、音频、CD-ROM 等）通信，而且每个子系统之间也能够用 DVI 总线通信。

82750 LH 主机接口门阵能够把 DVI 总线连接到主计算机总线，如能够连接到 PS/2 的微信号总线，也能够连接到 IBM PC/AT 的 ISA (Industry Standard Architecture) 总线。82750 LV 是第二个专用门阵芯片，它能够处理 VRAM/SCSI/Capture 接口功能，为数据从获取子系统和 CD-ROM 驱动器流向 VRAM 提供直接信道。该门阵还执行 VRAM 控制器的功能。第三个门阵是 82750 LA，它提供在 DVI 总线上到音频子系统芯片的逻辑接口，它还执行彩色键连和锁相的功能，使用它能够把 DVI 的视频信息彩色键连到其他的彩色图形源上，典型的例子是 VGA 卡。

II 型 DVI 系统 ActionMedia 750 II 的硬件体系结构如图 5-24 所示，分下面几个主要部分论述。

### 1. 视频子系统

视频子系统的关键技术是视频处理和显示引擎，它们由 82750 PB 像素处理器，82750 DB 显示处理器以及 VRAM 组成。存储器数组 VRAM 存放所有 DVI 系统数据，包括位映像数据，压缩编码数据，算法微码，控制执行算法的数据结构以及控制显示功能的寄存器数据。像素处理器 82750 PB 用微码执行视频图像快速处理算法，视频特技以及数字式运动图像和静止图像的压缩编码算法以及译码算法。显示处理器 82750 DB 是非常灵活的可编程的显示处理器，它能够将不同的位映像数据变换成在监视器显示需要的模拟的 RGB 信号。

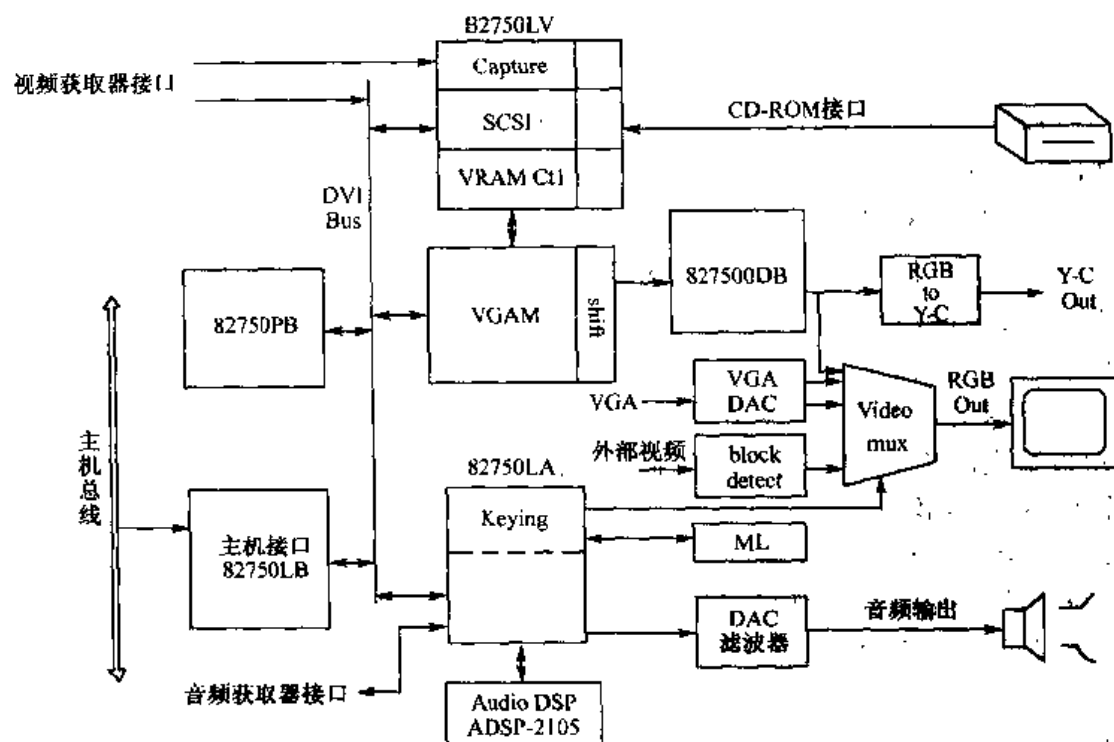


图 5-24 II 型 DVI 系统结构原理图

82750 PB 像素处理器直接连到 VRAM 的随机或并行信道。82750 PB 是具有较宽的指令字长（48 位）的快速微码处理器。在 25MHz 时钟时，它的运行速度是 25MIPS（Million Instructions Per Second）。由于指令字的不同字段分别控制不同的硬件机构，所以这些指令可以同时执行多种操作。因此，82750 PB 像素处理器每秒钟可以执行 100M 操作（100MOPS）。

某些指令字段控制的硬件结构特性特别适合于图像处理的运算操作，它包括两个分开并对称的内部 16 位数据总线，为 8 位像素计算专门分开的 ALU 操作；在解压缩时为运动补偿设计了像素插值器，为解释压缩编码数据流设计了统计译码器；以及为了同 DVI 的 VRAM 传输数据所设计的四个先进先出数据缓冲区（FIFO）。

一个典型的操作是 82750 PB 像素处理器运行较小的微码译码器，它定时询问在 VRAM 中的命令表。由主计算机建立命令表，微码命令由主机直接引导加载到 82750 PB 微码存储器中。当命令表指出某些操作需要运行时，例如译码操作，微码译码器从 VRAM 中将一个微码块加载到 82750 PB 内部的微码存储器中，并且执行它。这些微码是由主计算机设计并加载到 VRAM 中的。

一个微码块通常是一些循环指令，它将重复执行给定的算法。在 82750 PB 内部存储的微码允许以 25MIPS 速度执行。如果微码命令表是在 VRAM 中，要考虑大于 40ns 的存取延时时间。

在执行时，算法从 VRAM 中获取压缩编码数据、位映像数据以及参数数据，并把算法结果存到 VRAM 中。在解压缩运动视频图像时，压缩编码视频数据的几帧有效成份以及解压缩数据的几幅位映像需要同时存在 VRAM 中。压缩编码数据输入队列采用 FIFO 方案，它允许从外部设备（例如硬盘）传输数据速率的变化，也允许解压缩算法需要提供数据速率

的变化。解压缩的位映像队列也采用 FIFO 方案，它允许解压缩时，从一帧到另一帧速率的变化，即允许常用的每秒 30 帧速率的变化。它也支持运动压缩编码技术中使用差帧技术，即使用该技术能够用前一帧重构现在这一帧。

AVK (Audio/Video Kernel) 软件在 VRAM 中还开辟了当前的显示缓冲区，在这里视频图像能够复制或改变比例。用这种方法允许多个视频数据流以任意的窗口尺寸同时显示在屏幕上。在 82750 PB 和 VRAM 之间，所有数据传送要在 1 秒钟内完成 30 帧，有效的 82750 PB 和 VRAM 之间的接口增强了系统性能。82750 PB 非常紧密地连接到 VRAM 控制器，它是 VRAM/SCSI/Capture 门阵电路的一部分。该门阵可编程改变随机存储器的 RAS 和 CAS (行、列地址选通) 时序，以适应不同存储器芯片所需的时序特性。

根据存储器的情况，82750 PB 能够满足从 82750 DB 显示处理器对 VRAM 控制器提出的提供地址，提前传送数据以及刷新周期的需要。82750 PB 像素处理器有三种类型的数据传输周期的三组指针和间距寄存器，它们是亮度、色差和 82750 DB 寄存器组。82750 PB 像素处理器从 82750 DB 显示处理器得到 4 位 VBUS 信号，以满足这些数据传送周期的需要。一旦接受到请求，82750 PB 能够仲裁并把这个请求提前通知到存储器控制器，把相应的指针寄存器的内容送到 VRAM 地址总线，最后把间距增量加到指针中。

82750 PB 默认 VRAM 是视频引擎的一个重要成员，核心外的其他数据 (主计算机，CD-ROM，音频数据以及获取器数据) 都必须经过 VRAM 才能到达或送出 82750 PB。主机和 82750 PB 之间的主机接口门阵汇集和仲裁从各个不同数据源来的请求信息。

82750 DB 显示处理器如图 5-24 所示连到 VRAM 的串行或顺序信道。82750 DB 是一个非常灵活的可编程的显示处理器，运算时钟最高可达 45MHz，它能够支持各种不同的显示屏幕格式，包括 NTSC PAL，VGA 和 XGA。在最高时钟速度是 28MHz 时，不支持 XGA 和 PAL，其他都支持。因为 XGA 需要 44.9MHz，而为 28MHz 时系统软件 (AVK) 不支持 PAL。

82750 DB 显示处理器有几种不同的 VRAM 的位映像格式，可直接解释成在监视器显示屏幕上所需要的模拟信号数据流。除了常用的彩色索引和 RGB 格式外，它还支持亮度 (Y) 和色差 (U、V) 的格式，这种格式对于高分辨率的静态图像和运动视频特别有用。

对于运动视频，“9 bit YUV”格式是最常用的，在 VRAM 位射图中它用 8 位表示亮度，每 4 个水平像素和 4 个垂直扫描行用一个 8 位表示色差信号 U、V。YUV 数据在 VRAM 中分别在三个位映像区存储。换句话说，运动视频是用一个  $256 \times 240$  采样集表示亮度信号 Y 的位映像，用  $64 \times 60$  采样集表示一个 U 位映像和一个 V 位映像。

82750 DB 显示处理器把这三个位映像区取来并混合在一起，完成解压缩最后的一步。同时，通过计算水平和垂直方向每个 U 和 V 的四个采样点的平均值，完成色差信号的插值。一旦完成了插值运算，82750 DB 要进行从 YUV 到 RGB 彩色空间转换，把三个 8 位数字信号送到 D/A 变换器，最后输出 RGB 模拟信号到彩色监视器。

“9 bit YUV”显示格式需要 82750 DB 和 82750 PB 特殊处理，为了支持恰当的数据传送周期，它们要把 VRAM 中随机存放的数据变成顺序数据，因为亮度和色差数据对于给定的扫描线不是存放在相邻的存储空间，串行的 VRAM 信道可能包括亮度和色差数据，但是不在同一时间。解决问题的办法是：在扫描正程时实时接受高密的亮度 (Y) 数据，而在扫描逆程时接受低密的色差信号 (U、V)。

每一行 82750 DB 显示处理器最多请求 82750 PB 像素处理器两次数据传送周期, 82750 PB 紧密地连接到存储器控制器, 而且它有亮度和色差在 VRAM 中数据的地址指针。每一扫描行请求一次 Y 数据传送, 每四行请求一次 U、V 的数据传送, 因为它们的采样比为 4:1。

另外的数据传送形式称之为“寄存器型”传送, 每一帧至少一次。因为 82750 DB 显示处理器通过串行口要加载它自己的寄存器, 所以在消隐的每一帧开始时, 82750 DB 自动地建立寄存器传送请求。同样为了满足更高分辨率而改变帧的格式, 在帧内请求也可编程。

对于任何一种寄存器传送请求, 都存在影响 82750 DB 显示处理器编程的系统时序问题。像素处理器 82750 PB 和 VRAM 控制器对给定传送请求的响应时间是可变的。这就是说, 控制器可能正在忙于其他的正在进行的存储周期, 而且存储器它自己的存储周期的时间间隔也可能随着时钟速率在变化, 解决的办法就是编程控制器。插入请求之后, 对于数据传送周期, 为了完成 VRAM 串行信道的加载任务, 82750 DB 显示处理器必须等待。然后 82750 DB 把时钟数据加载到它自己的寄存器中。

82750 DB 中的像素控制寄存器的移位时钟延迟参数 (SCLK Delay) 必须编程反映 82750PB 和 VRAM 对它请求最快的反应时间。这个时间是下述几项的总和;

- 82750 PB 接受 VBUS 请求信号的同步时间。
- 当前使用的 VRAM 最长的存储周期。
- VRAM 传送数据的时间。

注意, 这些时间必须算到 82750 DB 的时间周期中, 因为 82750 PB 和 VRAM 控制器通常运行的时钟速率与 82750 DB 是不同的。增加一点点额外的溢出时间在这里是安全可靠的。如果在这里给的时间不够的话, 82750 DB 将接收错误的数据, 即在一个时钟周期内 VRAM 串行口的内容是错误的。这时系统将失败, 出现错误的亮度数据、错误的彩色数据或者错误的寄存器数据。由检查和检测出的错误的寄存器数据将引起系统停止。82750 DB 显示处理器发一个停机码 (Shut Down Code) 到视频总线 (VBUS)。

其他的时间问题也影响 82750 DB 的编程参数, 例如视频总线的请求在视频总线上保持的时间长度 (VBLEN) 以及 VRAM 串行时钟的速率等。时间问题决定在不同条件下 82750 DB 所能支持的最大的分辨率; 如水平消隐时间和 VRAM 串行口传输速度的平衡问题, 因为它能够决定在水平消隐时间有多少彩色数据传给 82750 DB 显示处理器。82750 DB 的时钟频率是很灵活的, 但是对于任何给定的像素时钟速率以及给定的 VRAM 速度时, 这种灵活性需要重新考虑。

## 2. 彩色键连子系统

II 型 DVI 系统 Action Media 750 II 通常认为在 IBM 主计算机上存在 VGA 或 XGA 图形系统, 它能够为 DVI 系统提供高分辨率的图形。VGA 和 XGA 是标准化的图形接口, 有很好的编程环境。这样就存在两个分开的帧缓冲区, 一个是为 DVI 运动视频的缓冲区, 一个是 VGA 或 XGA 高分辨率图形的帧缓冲区, 彩色键连子系统能够把这两个缓冲区结合在一起, 在屏幕上显示。

在单个帧缓冲区的情况下, 覆盖运动视频上的任何一个图形都需要以运动视频的帧速重新绘制这幅图形。这就是说运动视频帧每一次复制到显示窗口时, 高分辨率的图形也要复制到同一视频帧, 它包括先前的运动视频图像以及在运动视频图像上的图形。很清楚, 要完成这么复杂的事情, 需要更多的软件支持以及更大的 82750 PB 绘图带宽。

对于单个帧缓冲区，另一个复杂问题是运动视频的分辨率与高分辨率图形的匹配问题。DVI 运动视频标准位映像分辨率是  $256 \times 240$  像素（水平 $\times$ 垂直），对于由 CD-ROM 提供压缩编码数据速率，它是一个最佳图像质量和有效而合适的分辨率。82750 PB 能够提供足够的像素传输的带宽，可在一个任意尺寸内确定窗口的内部复制或改变位映像图像的尺寸。因此，如果把位映像分辨率扩大到  $640 \times 480$  像素（VGA 的分辨率），或者大于每秒 30 帧的位映像率，它的带宽会感到不足。如果 DVI 系统希望运动视频同时配上高分辨率的图形（VGA 或 XGA 产生的图形），就需要解决这个问题。

现有的 VGA 或 XGA 系统采用了将 DVI 系统模拟输出和上述图形系统的模拟输出相合并的办法，对于运动视频，它没有调用  $640 \times 480$  像素的能力，图像的结果仍旧是很好的。事实上 II 型 DVI 系统增加了灵活性，它能够把运动视频合并到任意主机图形系统的分辨率。用两种不同的方法，在水平方向上能够将 256 运动视频的采样增强到全屏幕宽：一种是 82750 DB 内部的方法，另一种是外部方法。

内部方法：通过编程 82750 DB 显示处理器选用每个像素大于 1 的时钟周期，能够增强像素（即采用 1.5, 2, 2.5 时钟周期/每个像素）。在水平方面上显示成人工的块状特性。更有用、更完美的内容技术是采用先进的 82750 DB 内插功能，在水平方向上将亮度信号扩展为 2:1。这就是说，对于位映像中的每个亮度信号，通过 82750 DB 显示处理器产生两个值。这些值的一个是位映像中的值，第二个值是邻近采样值的平均。彩色采样值的内插是用相同的方法，把水平方向的 256 个像素扩展成 512 个像素。

外部方法：通过压控振荡器产生 82750 DB 需要的时钟；由锁相回路将压控振荡器的振荡频率锁到与主机图形系统振荡频率一致。这样就能把 512 像素扩展到任意尺寸，在 VGA 的情况，640:512 扩展比（5:4）是需要的。键连（Keying）/音响（Audio）门阵寄存器控制锁相回路，能够建立产生这个长宽比，因此，在 20.140MHz DVI 512 像素系统上运行能够精确地与在 25.175MHz 640VGA 像素系统上运行相匹配。

这个技术解决了屏幕作用区的问题，但是当把 82750 DB 时钟系统锁相到外部的图形系统时，其他问题就表面化了。在锁相时，82750 DB 显示处理器的水平和垂直计数器由外部图形系统的水平和垂直同步信号置位到零。如果没有产生合适的水平的置位，没有保持相对于 82750 DB 的时钟特性，在一行到一行的扫描时就会出现水平方向的跳动。

无论如何，这种方法是一个先进可行的方法，它能解决一般情况下 DVI 显示系统与其他任意水平分辨率显示系统的匹配问题。为了提供较好的全屏幕的显示匹配，它不一定需要常用的 2:1 或 5:4 图像显示系统的长宽比。由 Intel 公司的软件直接支持标准显示格式的锁定，对于非标准的显示格式将需要开发 82750 DB 相应的寄存器集，并修改驱动器的原码。

注意，采用高频的 82750 DB（大于 45MHz）能够很容易将 DVI 显示分辨率匹配到高分辨率的图形系统上，如果从降低成本价格考虑，采用 28MHz 的 82750 DB 也能够匹配多种主机图形系统开发各种应用。

### 3. 音频子系统

同视频引擎并行操作的是音频信号处理器，数字到模拟的转换硬件以及模拟滤波器组成的音频子系统。它执行音频信号的压缩编码和解压缩，数字信号到模拟信号的转换，最后送到外部的音频放大器及音响系统进行播放。



音频子系统的核心是模拟设备公司生产的 AD 2105 数字信号处理器 (DSP)，通过它完成所有音频信号的压缩和解压缩任务。DVI 系统采用了自适应预测编码方法 (ADPCM) 算法把 16 位的采样数据压缩编码成 4 位码。DSP 芯片还能控制音量、采样速率变化，从 VRAM 中抽取压缩编码数据，最后将解压缩的音频数据输出到 D/A 转换器。上述操作 DSP 的编程码及查找表 (Look-up Table) 内容由主计算机处理器通过键连音频门阵电路加载。

像视频引擎一样，音频的数字信号处理器 AD2105 把 VRAM 作为它的压缩编码数据源，而且键连/音频门阵提供直接存储器存取方式 (DMA)，DSP 能够以 DMA 方式从 VRAM 获取数据，并把结果放到 VRAM 中。和视频引擎不同的是，在开始显示之前，音频 DSP 至少译码一或两帧位映像数据，译码之后音频 DSP 不需要把结果再返回到 VRAM 中。在它自己的 FIFO 中为输出数据只保存最短的队列。键连/音频门阵电路包括时序电路，在正常情况下每次新的采样输出时向 DSP 发出中断请求。在中断服务子程序中，DSP 将采样数据从 FIFO 队列中移开，通过门阵电路中的音频输出寄存器将它们传送到立体声的 D/A 变换器。音频输出寄存器完成 D/A 变换器接口需要的数据串行化任务。第二个 DSP 中断在获取音频数据进行压缩编码时，DSP 保持短的 FIFO，响应当门阵输入缓冲区满时从键连/音频门阵发出的中断请求信号。第三个 DSP 重要的中断是垂直消隐中断。这个中断每个显示帧出现一次，目的是解决视频数据流和音频数据流的同步问题。

在 DSP 和主计算机 CPU 之间，以及 DVI 任意设备之间的通信可能采用下述两种方法：一种方法是通过 VRAM 中的结构数据；另一种方法是通过在键连/音频门阵中提供的到 DVI 的信息寄存器 (MDVIR) 以及到 DSP 的信息寄存器 (MDSPR)。到 DVI 的信息寄存器提供从 DSP 到 DVI 主机及 DVI 设备的信息，DVI 设备是挂到 DVI 总线的子系统。当 DSP 把一个信息写到 DVI 寄存器，键连/音响门阵就会发出一个音频中断信号，通过主机接口门阵接上这个信号，主机接口门阵把这个中断信号经过主机扩展总线的中断请求送给主处理器。当主机或 DVI 设备把信息内容读到 DVI 寄存器后复位音频中断。

在主机或其他 DVI 设备写一个信息到 DSP 寄存器时，键连/音响门阵将门阵内部某一位置 1，DSP 每 1ms 定时查询这一位。如果信息出现，DSP 把这个信息读到 DSP 寄存器后，DSP 清除这位的标志。

数字到模拟量的转换器是由 Burr-Brown 公司生产的 PCM66P 单片立体声 16 位串行接口组成。跟着 D/A 变换器是双信道的模拟滤波器，它的截止频率近似固定在 17kHz，并且具有 5 个极点。音频采样输出的速率最高可达 44.1kHz，5 极点的模拟滤波器用 17kHz 上限的截止频率很好地消除了采样频率和图像频率的影响。由 DSP 能够完成上限采样的速率转换的计算，它支持几种不同音频数据流采样速率转换到 44.1kHz 输出速率。

#### 4. DVI 总线

为了支持视频和音频子系统，大量的基本数据必须在 DVI 的 VRAM 和 DVI 的其余设备之间传送，其余设备包括外部设备，主机以及获取子系统。在 DVI 系统中数据的通信信道采用具有多路开关功能的 32 位数据和地址总线，也称之为 DVI 总线。用该总线，不仅主机能够同 DVI 每个子系统通信（视频子系统，音频子系统，CD-ROM 等），而且子系统之间也能够用 DVI 总线通信。

DVI 总线是由 VRAM 并行信道的数据信号组成的，所有三个门阵，82750PB 像素处理器以及 VRAM 都直接连到总线上，很多时间，DVI 总线作为 VRAM 和 82750 PB 之间单一

的数据总线，它们是默认的 DVI 总线的主设备。有时 DVI 总线也要为 DVI 系统中其他设备传送数据，例如，当主机处理器需要读音响门阵寄存器的内容，在这种情况下，DVI 总线担当了不同的角色，它首先给出 DVI 设备的地址，然后再传送有用的数据。

为了在 DVI 总线上传输数据，首先必须把 DVI 总线控制权从 82750 PB 手中转让给申请控制权的 DVI 设备，主机接口门阵作为各种请求的仲裁器，这些请求可能来自主计算机音频子系统，CD-ROM 子系统或者获取子系统。CD-ROM 和获取子系统的请求通过 VRAM/SCSI/Capture 门阵合成一个信号。仲裁器采用按时间先后极其公正的方法，然后逐一执行。

一旦一个请求信号被仲裁器承认了，总线控制权从 82750 PB 手中转让给该设备，允许在 DVI 总线执行该设备的通信协议。当 DVI 总线可用时，主机接口门阵将从 82750 PB 得到主机总线允许信号 HBUSEN (Host Bus Enable Signal)。主机接口门阵给请求设备发一个回答信息、AUDACK 或 CACK 信号。这时，请求设备发地址到 DVI 总线。主接口门阵同时还要观察 82750 PB 的有效地址允许信息 (Valid Address Enable Signal)，一旦这个信号建立起来，主接口门阵将不接受其他的设备请求，同时通知请求设备，DVI 总线已从地址总线切换到数据总线。

因为 DVI 总线实际上是由 82750 PB 的数据线组成的，当 82750 PB 需要存取其他 DVI 设备的数据时，问题就出现了，即具有多路开关的地址/数据总线的方法不能很好地解决问题，因为 82750 PB 是总线的主设备，而且把 DVI 总线只用作数据线。在这种情况下，选择几条 82750 PB 的地址直接连到几个门阵电路，将 VA19:17 地址线连到所有门阵电路，用来获得 DVI 设备的 ID 信息，低位地址线用来选择 DVI 设备的内部寄存器。

## 5. 获取子系统

获取子系统通过 DVI 总线和信息交换控制线协议与外部数字化设备相连。在 VRAM/SCSI/Capture 内部是地址指针和计数器，用它把从数字化仪得到的数据送到 VRAM 中。数据流是通过 VRAM 的随机信道。这种方法相对于串行信道会受到存储器存取速率带宽的限制，连接方便，造价比较低。如果将获取的数据通过 VRAM 的并行随机信道存到存储器，VRAM 的串行输出信道就能立即把它送到监视器，显示帧存储器的内容。

采用并行信道方法的另一个好处是获取系统不需要锁相到显示系统的频率，甚至较慢的扫描仪的速度都可以。获取系统也能够从隔行电视扫描光栅源获取数据，如 NTSC 或 PAL，显示输出的图像照样可以选用非隔行扫描的设备，如 VGA 或 XGA。获取系统还能够获取 82750 DB 不支持的位映像形式，如 YUV12。

II 型 DVI 系统 ActionMedia 750 II 获取器模块是 DVI 系统 Delivery 板的一块子板，即作为 Delivery 板的一块背板和 Delivery 板一起占用一个 IBM PC 机的插槽。现在获取器模块设计是非卖品、还不能仿制生产，但是 Intel 公司已经在 ActionMedia 750 II 技术参考手册中公布了获取器模块总线接口技术规定，为希望自己设计获取模块的用户提供方便。

## 6. CD-ROM 子系统

VRAM/SCSI/Capture 门阵包括了一个小型计算机系统接口 SCSI (Small Computer System Interface)，这个接口用于支持单个 CD-ROM 驱动器，作为压缩编码视频和音频数据源。把这个子系统加到 II 型 DVI 系统中，不需要外部扩展卡，用较低的成本就能支持 CD-ROM 驱动器。另一个好处，从 CD-ROM 驱动器读出数据到 VRAM，采用上述方法，可以减少对

扩展总线带宽的需要。

在 CD-ROM 子系统中, 采用了灵活的链块的直接存储器存取方式 (Chain-block Direct Memory Access), 同时, 提供一个语法解释器的工具, 即把到来的压缩编码的数据流解释成连续的视频和音频成份, 并把它分别放到 VRAM 的不同区域中。Intel 公司的 AVSS2.20 全面地使用了这个特性, 它需要这个语法解释器特性。Intel 公司 AVK 软件执行数据流扫描方法, 它把到来的数据流放到 VRAM 连续的空间中。

7. 主机接口

主机接口门阵为计算机系统监视并控制 DVI 系统提供多种方法。该门阵通过选择不同的门阵管脚能够连接到 IBM PC 的 ISA 总线标准或者 PS/II 的微通道 (Microchannel) 总线标准。

VRAM 能够通过扩展存储器空间 (Expanded Memory Space——EMS) 窗口, 或者经过 I/O 映像先进先出寄存器 (First In First Out——FIFO) 存取 VRAM 中的数据。上述方法具有自动加 1 的特性。

能够通过 EMS 窗口, 或者通过使用 EMS 窗口的快速方法存取 DVI 设备寄存器的内容, 同时, 它还提供单周期修改 DVI 设备地址指针的方法。

除了支持综合的数据 I/O 外, 主接口门阵还可作为从 DVI 子系统到主机扩展总线中断的仲裁中心, 中断源可能是视频子系统 (82750 PB)、音频子系统、CD-ROM 子系统以及获取器子系统。依据屏蔽和把这些中断源映像到变化的扩展总线的中断级, 可编程现有的中断系统。

5.2.2 DVI 软件系统中的 AVSS

1. DVI 软件系统概述

第一代的 DVI 系统软件如图 5-25 所示, 图 5-25 给出了 DVI 运行软件的所有模块及它们的相互关系。图中最下层是 DVI 系统的硬件。较早的一型 DVI 硬件系统包括 DVI 视频板, 音响板, 多功能板以及 IBM PC/AT 的硬件。

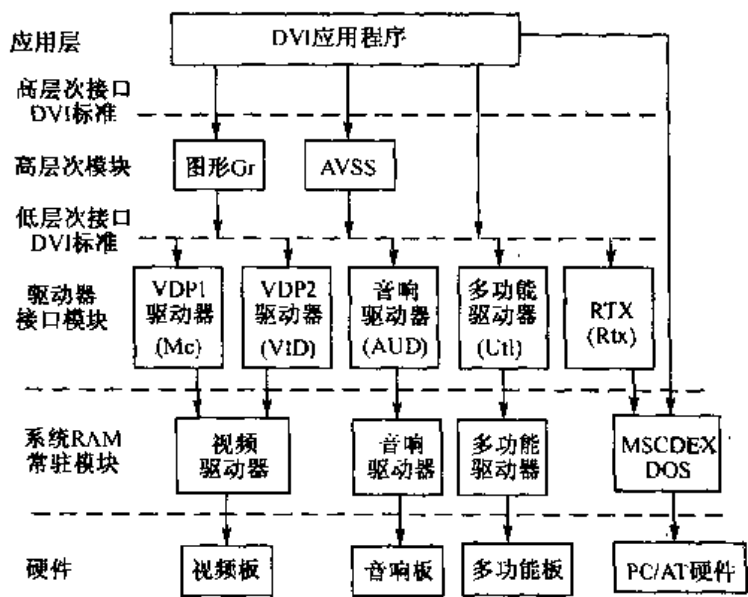


图 5-25 DVI 运行软件结构图

再上一层是驱动接口模块，驱动接口模块建立了为高层应用软件所使用的虚拟设备。驱动接口的调用是应用程序最低层的存取功能。连接驱动程序的是驱动程序接口模块，在 DVI 系统中共有四个驱动器接口模块。

(1) 微码接口模块[Mc]。它是 82750 PA 的接口模块，负责微码的加载和执行，同时也负责主机系统对 VRAM 的存取。

(2) 视频接口模块[Vid]。它是 82750 DA 的接口模块，负责 82750 DA 的初始化。同时，它还包含了视频信号数字化器的接口软件。

(3) 多功能接口模块[Ut1]。它提供 CD-ROM 和操纵杆的接口软件。

(4) 音响接口模块。它是音响板和音响数字化器的接口软件。

在同一层次上还有两个 IBM PC/DOS 的扩展模块。

(1) 实时执行模块[Rtx]。它为 DVI 应用软件提供实时多任务操作环境。

(2) Microsoft CD-ROM 扩展模块[MSCDEX]。它是 DOS 扩展模块，它能够使满足 ISO 9660 的 CD-ROM 文件用一般的方法在硬盘和软件上自由存取。

在 DVI 软件系统中还有两个高层次的软件包，如图 5-25 所示一个是图形软件包[Gr]，一个是音频视频支撑软件 AVSS[Av]。

(1) 图形软件包[Gr]。它提供图像处理，图形绘图基元以及视频管理功能。图形软件包是一组管理位映像信息的子程序。图形子程序可以用结构图形、图像和文本三种不同形式进行操作。

(2) 音频视频支撑系统 AVSS[Av]。AVSS 软件可管理用 AVSS 格式写的视频、音频文件。它也能播放从 CD-ROM、硬盘或 RAM 中读出的视频、音频数据。AVSS 文件格式比一般的文件复杂得多，它支持多数据流，例如，它可包括一个视频数据及两个立体声的音响数据，能够较好地解决视频信息和音频信息的同步问题。

最上面是应用层，它可以提供大量的应用程序，例如导游、销售点、教育培训以及文文件管理系统等等。在应用层下面还有两个高层的 DVI 系统的接口。

(1) DVI 系统生产工具软件。它能够获取、压缩和显示静态图像，同时能完成静态图像各种格式的转换。它能够实时获取、压缩运动的视频图像和音频数据，建立、复制、编辑以及播放 AVSS 音频和视频文件，记录、编辑和播放音响文件。完成上述功能它不需要用 C 语言编程，它能够在 DOS 提示符下解释执行各种工具命令，而不需要像 C 语言那样的编译和连接。

(2) 多媒体编程工具著作语言。用著作语言编制应用程序，它能够把应用程序构造成流程图或者故事板 (Story board) 的形式。它能够用简单的语言描述所有不同的屏幕和序列，然后著作软件建立应用码，同时允许开发者根据经验交互式地改善和修改应用程序运行结果。上述过程结束，应用程序就生产出来了，应用者可以完全不了解 C 语言和 DVI 系统的详细情况。

## 2. AVSS 模块

播放 DVI 系统中运动的视频和音频数据使用了音频视频支撑系统软件子程序模块，简称 AVSS (Audio/Video Support System)。AVSS 需要视频和音频特殊格式的数据，叫做 AVSS 文件格式。下面将 AVSS 操作的基本概念作简单的介绍。例如，AVSS 播放运动视频，运动视频加上音频，单独的音频以及静态图像。AVSS 也允许应用程序同步播放视频数据。

(1) AVSS 文件格式。AVSS 文件格式(扩展名为.avss)能够提供较多的灵活性和扩展性。它比图像文件更复杂,它能够支撑多数据流同时操作,例如一个 AVSS 文件可以包括一个视频数据流和两个立体声音频数据流。更复杂的情况,一个 AVSS 文件可以包括 4 个视频数据流和 4 个音频数据流,以一个多窗口的视听效果播放,或者是应用程序希望播放 4 个视频数据流中的一个,需要时立即切换到另一个。

每个文件多个数据流的方法在多媒体软件系统中是非常需要的,当使用较慢的存储设备时,如 CD-ROM,一次能够连续播放运动的视听数据一定要采用这种方案。AVSS 系统曾经尝试过用并行的方案同时播放两个不同的 AVSS 文件,这需要较大的缓存和较快的存储设备。

根据音频和视频流的类型,AVSS 文件又提供了三种附加数据流的类型:底层数据,数据和图像。底层数据是任意一种类型的数据,它必须同视频和音频信息一起实时提供,例如一种底层数据是时间码,它与 AVSS 文件的每一帧一起存储。数据流是由计算机常用的数据序列组成,直到全部数据流存储到存储器之前,应用程序不能使用该数据。AVSS 系统有责任把视频、音频和图像流转换成声音和图像。AVSS 系统允许应用程序使用底层数据,并为其提供方便,例如当底层数据到达时,存取底层数据;从数据流收集底层数据并把它存到由应用程序规定的缓存区中;当数据流到达时,通知应用程序。

保存 AVSS 文件的介质,如 CD-ROM,硬盘的 RAM 盘,只能使用二进制代码的单数据流。因此在 AVSS 文件中的多数据流只能采用如图 5-26 所示方法将多数据流交叉变成单数据流文件,这是 AVSS 文件最基本的格式。AVSS 文件可能是相同的视频帧信息,也可能不是。它们是一束数据,该数据包含一块从有用的数据流开始的完整的数据。通常在 AVSS 帧中一个视频块应该是单个图像或一个定义完整的视频帧(虽然视频帧可能是一个帧到一帧压缩码数据,但是它假设提供的帧数据是可用的),这个文件格式也能够表示不完整的图像的一个视频块的数据。

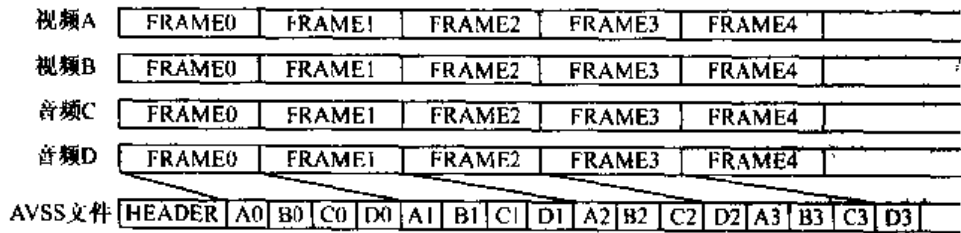


图 5-26 多数流交叉变成单数据流文件

AVSS 一帧单独的一块也可能没有固定的尺寸,特别是经过压缩编码的视频数据,某些帧可能比另外一些帧多些或少些数据。播放时要保持一定的平均速率,对于慢速的存储设备是非常重要的。要播放一个长序列数据必须克服 CD-ROM 需要的寻找时间,一圈的寻找时间至少要引进几百毫秒的延迟,也可能更多,或许它可能引进不能接受的数据中断。因此,AVSS 文件常常包含一些缓冲衬垫数据,为了控制平均字节速率,通常附加一些空位的数据流,与其他有用的数据流交织在一起。

(2) AVSS 文件的操作。通过 DVI 系统中的实时执行模块 RTX 操作 AVSS 文件。为了说明 AVSS,我们将引进一些 RTX 的概念。RTX 是运行 DVI 应用程序中的多任务形式。RTX 的基本工作单元是任务;建立 PC AT 的中断结构,因此,很多 RTX 任务能够“同时”

操作。具体是每秒钟中断 80286 CPU 30 次来实现上述的“同时”。应用程序中任务的执行取决于中断优先排队结构，RTX 调度将给每个任务顺序运行的机会。各个任务通过系统事件标志能够完成相互的通信。

当运行 AVSS 文件时，它将激活三个优先 RTX 任务：几个从外存送到 RAM 的数据传送任务；VDP1 任务，处理控制像素处理器 82750 PA 的任务；VDP2 任务，处理图像显示任务。这三个任务完成了播放视频和音频全部工作。

图 5-27 给出了与 AVSS 有关的数据流方框图，在图的左下角读入数据，从硬盘，CD-ROM 或 RAM 读入数据。第一步是数据的语法分析，如前所述，把数据分解成单个的数据流：视频数据送到 VRAM，音频数据送到音频板的 RAM 中，底层数据送到应用程序在系统 DRAM 中建立的缓冲区。图 5-27 给出 AVSS 用 VRAM 所做的工作，输入缓冲区 VRAM 存放经过压缩编码的视频位数据流。“位数据流”、“显示表”是存放 VDP1 (82750 PA) 显示命令表的 VRAM 区，图形软件模块使用视频的空间，通常在 VRAM 顶部的位置。“DB<sub>n</sub>”等是一些显示缓存区，用来解压缩视频帧数据。显示缓存区典型数据是六个，当然有些情况需要更多一些。AVSS 的 VDP1 任务负责通知 VDP1 解压缩位数据流，并且把结果放到图上特殊的缓冲区，解压缩数据缓冲区 DB<sub>n+1</sub>。当然，VDP1 也可能存取前一帧数据 DB<sub>n</sub>，同时通过显示表得到操作命令。如图所示，在同一时间，VDP2 任务告诉 VDP2 (显示处理器 82750 DA) 显示再上一帧 DB<sub>n-1</sub> 中已经完成了解压缩的数据。因此，在当前显示使用的一个缓冲区之前，已经有几个缓冲区正在使用。

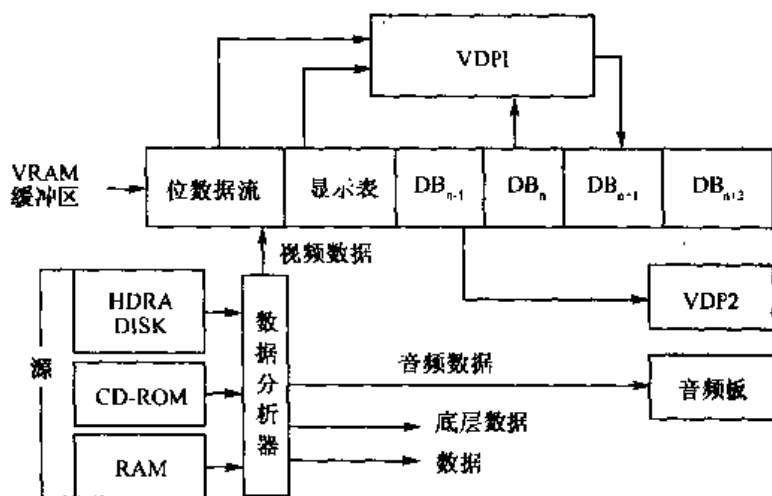


图 5-27 AVSS 数据流图

当 AVSS 文件正在运行时应用程序要做其他事情是可能的，这是因为 AVSS 中有一个 Hook 子程序的概念。Hook 子程序是一种 C 语言功能，应用程序能够为每个视频帧在 AVSS 进程的特殊点建立 Hook 子程序调用。图 5-28 给出了在 AVSS 流水处理时，三个 Hook 子程序调用点。图中从左向右也说明视频帧数据在系统中的处理过程。三个方块表示语法分析、解压缩以及显示激活的三个 AVSS 任务。在任何时间，每个任务的前面都要把某些东西存到缓冲区中。Hook 子程序能够在图上 Hook 符号表示的点建立 Hook 子程序的调用。当任何一帧压缩编码数据在压缩编码缓冲区完成压缩编码并且开始解压缩处理时，第一个 Hook (0) 能够被调用。如果 Hook 子程序在这里激活，它标明了子程序将被调用、通过的帧数以及帧

缓冲区的位置。

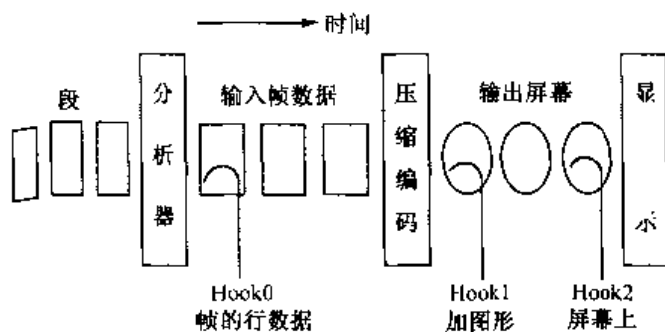


图 5-28 AVSS 流程图和 Hook 子程序位置

第二个 Hook 子程序调用点 Hook (1) 是在帧解压缩之后的瞬间。通常这点是在缓冲区中输入绘制的图形。最后一个 Hook 子程序调用点 Hook (2) 是在帧数据将要显示之前的瞬间。在这一点 Hook 子程序显示一些与帧紧密相关的事情，例如根据鼠标在屏幕上画光标。

由 Hook 子程序使用的 CPU 周期或 VDP1 周期必须注意统一规划。通常 VDP1 全部由解压缩任务占用，除非解压缩任务采用短时译码。Hook 子程序希望扩大 VDP1 的处理任务，对于压缩编码的服务必须更进一步减少解压缩的时间。因为减少解压缩时间意味着输出图像质量的下降，所以必须细心选择确实可接受的图像。另一个考虑是，Hook 子程序产生的 VDP1 显示命令表，要和 AVSS 的 VDP1 任务产生的显示命令表相结合，关键问题要考虑 Hook 点的位置，除非在发出 VDP1 任务前 Hook 子程序占有大量的处理时间，一般将没有什么问题。

在播放 AVSS 文件时，80286 CPU 通常不是满负荷工作，它有很多时间周期可用来完成其他任务。但是相反，在播放视频和音频任务时，要计划大量的使用 CPU 时间。为确保合适的操作，必须仔细地估算和实验。

(3) 附加的 AVSS 特性和能力。AVSS 有很多默认的参数值，如播放帧的速率，在屏幕上视频信号的位置，音频的速率、音量，视频音频相关的时序等。在播放之前和播放的瞬间，上述参数都能改变。通过 Hook 子程序在底层数据流包含的信息上，改变上述众多参数是可能的。

如果有足够多的 CPU 周期和足够多的 VDP1 时间，就能够在同一时间运行两个或多个 AVSS 文件。如果文件都存在 CD-ROM 上，需要寻找时间，上述问题是不行的；但是如果把文件存在不同的存储设备中，上述问题是可能的。例如，在导游的介绍应用中，AVSS 播放的背景音乐是从 RAM 盘中调入，播放的视频数据是从 CD-ROM 中调入的。RAM 盘中的音响数据作为单独的音响数据文件或者是伴随导游介绍视频信息，较早地从 CD-ROM 中读到 RAM 盘中。要在同一时间播放视频流，另一种办法是一个视频数据流来自 CD-ROM，另一个视频数据流来自硬盘。如果运行多个 AVSS 视频处理，则必须注意要为每个视频处理提供充足的缓冲区。

AVSS 是为播放运动视频 DVI 而设计的软件。正如前面所说，我们称 VDP 芯片组 (82750 PA 和 82750 DA) 是 DVI 硬件系统的核心，在这里我们应该称 AVSS 是 DVI 软件系统的核心。AVSS 是演示 DVI 技术中所有视频和音频计算机能力的软件，它允许应用设计者在同

一时间单独或混合使用 DVI 技术中所有的演示模式。

### 5.2.3 Windows 系统环境下开发的 AVK

#### 1. 新的概念模型

对于下一代 DVI 系统软件，新的模型已经研制开发出来：数字视频信号生产演播器。一个典型的生产演播器应包括混合器，磁带，监视系统，特技处理器以及为了记录、修改和播放视频和音频信息联在一起的其他设备。在我们的概念模型中，数字式生产播放器是控制管理数字式音频和视频数据流子系统的集合。数据流试图集中一起播放，形成一个“组”。模拟量的输入和输出可以想像成一个“信道”。音频和视频数据流从输入信道通过连接到输出信道。数据流能够混合在一起，例如输出到同一个输入信道，或者它们能够通过一个“效果处理器”用同样方法交换数据。如图 5-29 所示，数字式生产演播器主要的组成部分是模拟设备接口，显示管理器，采样器，效果处理器以及音频/视频混合器。

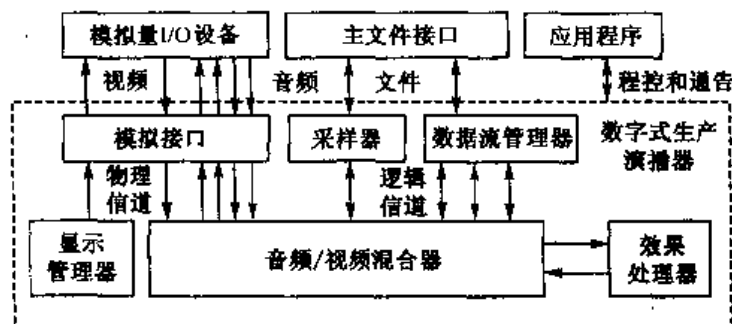


图 5-29 多媒体数字式生产演播器的概念模型

(1) 模拟接口。模拟接口允许系统连接到各种模拟的输入设备和输出设备，例如，激光视频盘，VCR，摄像机以及话筒和扩音器。把这些连接叫做物理信道，因为它们直接映像到外部实际的设备。一个实际信道不是双向的，它们的作用不是输入就是输出信号数据流。这些连接实际上是对应 DVI 系统硬件的输入输出特性。DVI 系统 II 型产品叫做 ActionMedia II，它的硬件允许视频输出到计算机的监视器、Super-VHS 录像机、电视监视器以及立体声的音响输出。视频和音频的输入组成在一起，连接到一个信号插头上。在所有的输出信道都要完成 D/A 变换，在所有的输入信道都要完成 A/D 变换。

(2) 显示系统。显示子系统控制在计算机屏幕上可视数据的显示。多种可视数据类型（静态图像和运动图像）可同时显示在相同的屏幕上。这些集合可以想像成用户的“观察窗口”数据，系统可以定义多于一个“观察窗口”。显示子系统能够建立这些窗口，并且在任意瞬间能够选择并显示这些窗口。这就是具有多个子混合器的等效视频信号，在某一瞬间只能监视这些视频信息的一个。

(3) 采样器。采样合成器或采样器是音响生产演播器中通用的附件。音响采样记录音响源所建立的波形作为数字式的序列存在存储器中。通过控制这些值能够改变音响效果。这个概念也能扩展到视频数据。一帧数字化的照片可以想像成运动图像的采样，显示之前在存储器能够进行控制和管理。

在我们的模型中，采样作为视频存储器块执行，该块可以用来暂时存放视频和音频的数据。从视频文件中能够采样一帧数据，并且可以作为静态图像存储。一幅图像可以送到效果



处理器，并传送到其他部件，在传送时能够改变色调、亮度等视频属性，同时进行转换，从一种位映像格式转换成另一种。然后放回到采样器并且显示或者存到盘上。

(4) 数据流控制器。数据流控制器特别像磁带机传输控制器负责控制数字式数据流，读写存储设备，例如 CD-ROM、磁盘或者局域网。数据流流过逻辑信道就像磁带的磁道一样，和物理信道的数据流不一样，逻辑信道的数据流不需要一对一的映像到物理设备上。例如 CD-ROM 设备能够把交错的视频和音频信号文件提供给数据流控制器，数据流控制器能够把它们映像到几个逻辑信道中去（一个视频信道和两个音频信道）。除了 CD-ROM 之外，逻辑信道是双向的。这就是说，不在同一时间、同一个信道，既可作为输入，又可作为输出。

(5) 效果处理器。效果处理器能够用来增加图像的特技效果，在静态和运动图像上增加图形，或者改变音响数据的特性。效果处理器作为特技效果单元，执行与 AVSS VCR 模型一样的功能。作为 VCR 播放时，它能够用来增加图像的特技效果，并给静态图像和运动图像增加图形。

(6) 混合器。音频和视频混合器是生产演播器的核心。它为将要到来的数据流提供分配到输出信道的机制，或者相反。例如，数字化左和右两个信道的音频数据，然后将它们组合成单一立体声的数据流，视频信道也能够提供连续的运动的数据流或静态的图像。

当一个音频输入信道产生单个的音响数据流时，混合器能够把多个音响数据流用多种方法混合在一起送到音频的输出信道。混合器能够用多种方法控制数据流通过系统。录音机或音响演播器通过从左向右滑动电位计能够使音响数据从左边输出信道向右边输出信道漫游。数字式音响数据流能够用各种百分比，范围从 0 到 100 分配给左边和右边输出信道。

混合视频数据流到视频输出信道是简单的定义源实体（运动图像或静止图像）以及目的实体（在计算机屏幕上应用者的窗口）的问题。几个源能够分配到相同的目的实体，产生复合的图像或者是“可视性的混合”。混合器实际上为修改数据提供某些控制功能。在数字式视频的情况下，它包括定义源和目的实体的矩形区，可用于剪裁，改变尺寸，变换属性，如色调、对比度、亮度以及色饱和度。

(7) 带宽饱和。在音响生产演播器中，如果太多的输入信道分配到磁带的同一磁道上，磁带会产生饱和现象，它将严重的影响声音的质量。同样，能够通过计算机实时解决大量数据流和数据流组存在的各种实际限制。例如，各种设备的数据传输率，主机总线的带宽，可用的 CPU 周期的数量，所有这些都影响计算机处理这些复杂的数据类型各种可能的限制。这些限制在数字式生产演播器模型中用饱和这个词来描述。

逻辑磁道可能是这样分配的：它们超过了某些设备的数据传输率，例如 CD-ROM 或硬盘。结果可能丢失记录数据，或者可能不合适地播放文件。在这两种情况，都可以说成是饱和。

当太多的特技效果同时安排播放时，将出现另一种形式的饱和。视频和音频处理器也受带宽的限制，当视频播放时，分给视频音频处理器的任务超过了它们能够实时完成的范围，结果就会出现丢帧或使声音速度放慢以及产生很多噪声。

另一个限制因素是总线的带宽，即总线数据传输率的限制。在 DVI 系统执行解压缩编码和显示之前，数字式数据必须从存储介质或局域网传送到 DVI 的局部存储器中。如果几个实际设备的数据同时传给混合器，这时系统总线的带宽会达到饱和，以至于不能传输

而且在这里把它放在 AVK 结构的外面，目的是为了更方便地移植到其他支撑环境。由 Microsoft 公司扩展的多媒体 Windows 系统（Multimedia Extensions to Windows）定义的媒体控制接口 MCI（Media Control Interface）以及 QMT（Quicktime Movie Tool box）接口都是特定环境接口的例子，它们都能在 AVK 上层执行。

3. AVK 中的实体

使用上述的概念模型，能够标识出众多实体并抽象出基本的 AVK 接口。接口是由实体及其相关的行为和属性的集合组成。图 5-31 给出 AVK 的实体和它们之间的关系。

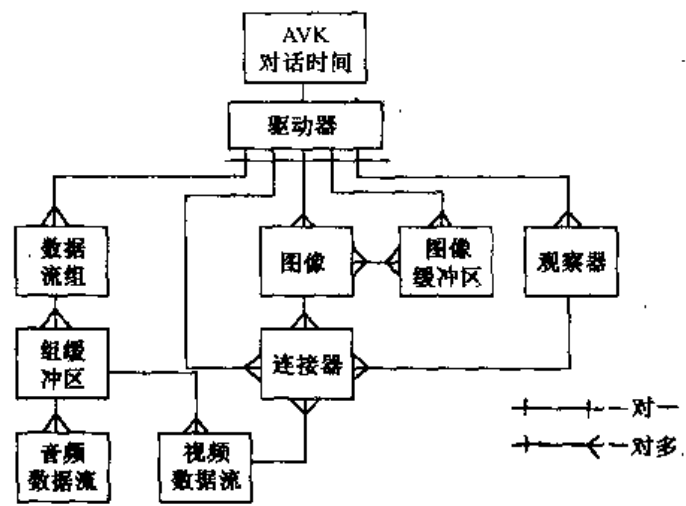


图 5-31 AVK 实体之间的关系图

模拟接口子系统由两个实体组成：AVK 的对话时间和 AVK 的设备。和这些实体有关的调用使 AVK 开始工作，定义 AVK 和应用程序之间的通信机制，允许询问设备的能力以及 AVK 应用程序运行时间，打开和关闭 DVI 设备。

数据流控制器可以看成是控制数字式数据流实体的集合。AVSS 把这些实体当成紧密耦合文件的集合，而 AVK 把它们当成独立的存储数据的文件来处理。数据流组是控制同步和通信的单元，如磁带的传输控制器。数据流组的调用可执行播放、暂停以及记录等功能。数据流组的缓冲区相当于磁带，它可能包含用相同速率播放的多个数据流。数据流是音响或视频单个的数据轨迹，几个数据流能够存储在相同的文件中。

在 AVK 中混合器的功能是由叫做连接器的实体提供的。连接器可以想像成是个流水线，它允许任选的传送数据流入和数据流出。连接器是复制功能高层次的抽象，连接为源和目的的位映像定义叫做“盒子”的矩形区。为了支持窗口需要改变图像的比例和重新定位，因此“盒子”的尺寸能够实时地修改。

在实体中把显示子系统具体叫做“观察窗”，观察窗显示图像，而且是把应用程序映像到窗口的可视区域。观察窗常常是连接器的目的，在这里源是另外的可显示的实体，如图像或可视的数据流。应用程序能够在多个观察窗之间切换选择。

现在的设计，采样器仅由图像和图像缓冲区实体组成。图像是可能存储静态图像 VRAM 的一部分。图像缓冲区是压缩编码的图像，图像可能被压缩编码到图像缓冲区，而且图像缓冲区能够被压缩成一幅图像。在替代一幅图像时，或者使用连接器进行复制操作时，都能够执行图像特技操作。

#### 4. AVK 的数据流

由于性能指针的原因 AVSS 选择使用了显示缓冲区的屏幕数组位映像技术。以前的 A 型 82750 PA 像素处理器对于实时以每秒 30 帧的速度解压缩  $256 \times 240$  像素的图像具有足够的的能力,但是要很快地复制和改变图像比例尺,A 型的处理速度就显得不够了。新的 B 型 82750 PB 像素处理器比 A 型处理速度快二倍多。利用增加的能力,能够用来更灵活地调度多个缓冲区,图 5-32 给出了 AVK 数据流程图。

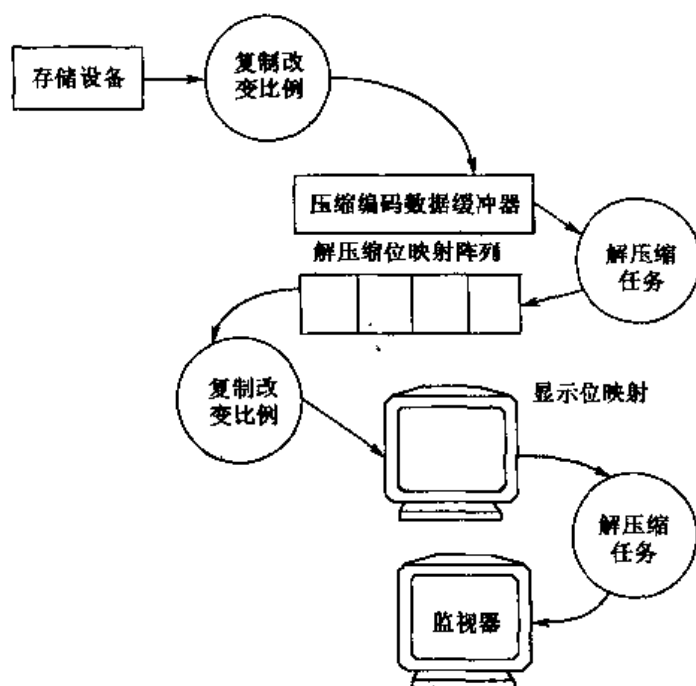


图 5-32 AVK 数据流程图

从解压缩位映射数组分离显示的位映像允许插入复制和改变比例尺的操作。它也允许窗口的视频效果,例如,重新定位和重新改变视频图像的尺寸。使用 AVSS,为了使视频图像出现在四分之一屏幕窗口,位映像全都需要  $512 \times 480$  像素的分辨率,把它分成四个区域,解压缩的位映像将是  $512 \times 240$  个像素,仅用一个  $512 \times 480$  像素位映像作为显示缓冲区。因为这有四个最小的位映像需要解压缩,这种方式实际上是使用最少的存储器。

AVK 数据流另一个优点是:由于 DVI 硬件具有更多的功能,多个视频窗口能够同时显示在屏幕上。为了完成上述功能,当对于相同的显示位映像数据执行复制和变换比例尺操作时,为每个视频数据流定位压缩编码数据缓冲区和解压缩数组数据。而 AVSS 结构是使用单个位映像数组用于解压缩和显示。

#### 5. 使用 82750 PB 像素处理器的任务调度

播放和记录数字式运动的视频和音频数据需要实时处理几个相互关联的任务。台式计算机中使用的很多种操作系统都不能对上述任务提供支持。采用了具有良好的可编程性的 82750 PB 像素处理器芯片为 AVK 解决了这个困境。82750 PB 使用叫做微码引擎的微码子程序集合执行实时的任务调度解决了上述问题。图 5-33 微码引擎原理图给出了微码引擎的组成部分以及它们的相互关系。主要的组成部分是调度器 (DoMotion),缓冲区/数据流处理任务,命令表处理任务以及周期的处理任务。

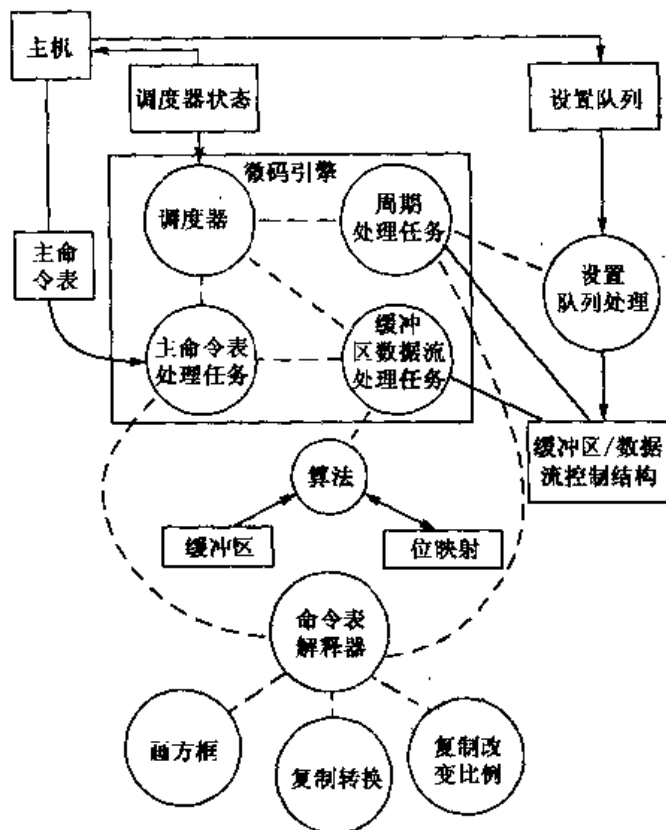


图 5-33 AVK 微码引擎

当视频数据流正在播放时，DoMotion 在主命令表处理任务和缓冲区/数据流处理任务之间连续循环。回放时，缓冲区/数据流处理任务为帧的压缩编码数据变成可用的数据进行循环处理，同时把位映像变成能够解压缩的帧数据。当这两个条件都满足时，称它为合适的解压缩算法。在获取视频信号的时候，缓冲区/数据流处理任务循环执行。为了从数字化设备得到位映像数据，把压缩编码的数据放到缓冲器中；然后调用压缩编码的算法。

视频信号处理的很多算法都在 82750 PB 像素处理器上执行。现在有两种类型的算法：PLV 和 RTV 算法。PLV (Production Level Video) 叫做生产级视频算法，它是脱机视频压缩编码算法，它用并行处理计算机而且提供最好的图像质量。PLV 能够用 892750 PB 实时地实现解压缩算法。RTV (Real Time Video) 为实时的视频，使用 82750PB 像素处理器实时地执行压缩编码算法和解压缩算法。

对于静态图像还有很多其他种类的压缩编码和解压缩算法，16 位和 9 位图像 DVI 技术的压缩编码和解压缩算法以及 JPEG (Joint Photographic Experts Group) 算法。

声音压缩算法是由声音的数字信号处理器——DSP 完成的。AVK 具有的算法包含在 ActionMedia 系统中使用的 ADPCM4E (a 4-bit Adaptive Defferential Pulse Code Modulation) 算法及 Microsoft 公司在多媒体扩展 Windows 系统所使用的 8 位 PCM (Puose Code Medulation) 算法。

主命令表处理任务执行由主机经过叫做队列集存储区请求的微码功能。命令表解释器使用主命令表作为指令堆栈。主机通过把命令放到队列集中能够动态编程 82750 PB 像素处理器。在周期处理时，把这些命令重新定位到命令表中。命令表提供每个微码功能的地址以及

它执行时间的参数。这些微码函数必须存放在设备存储器中，因此能够把它实时地加载到 82750PB 像素处理器在芯片上的指令 RAM 中。当从周期处理任务调用命令解释器时，它将从命令表数据流中得到它的指令，当从主命令表处理任务中调用它时，它将从主命令表中得到这些指令。顺序执行每条指令，当执行完成后返回到解释器。当整个命令表都执行完的时候，命令表解释器返回重新调用任务。

场消隐中断触发了周期处理任务的执行，如果执行复制一帧的操作，它将通过命令表解释器调用复制和改变比例尺的微码。在这种情况下，使用命令表数据流代替主命令表。

复制和改变比例尺的微码能够复制一个矩形区从一个位映像到另一个位映像。它能够任意改变映像的比例，或者通过改变色调，对比度、饱和度和亮度，变化图像显示的状态。如果位映像是一串视频帧之一，复制作为解压缩帧数据连续执行。如果位映像是静态图像数据，复制只是在需要时执行一次。通过应用程控复制和变化比例尺的操作，使用 AVK 接口连接器实体。

周期处理任务调用队列集函数，并且用队列集函数从主机传送命令表或命令表数据流。这样就允许主机传送指令到 82750 PB 像素处理器。AVK 使用 82750 PB 作为协处理器或者从处理器（和 AVSS/RTX 系统相似）。由 DoMotion 执行实时任务调度，使主机 CPU 有更多的时间执行数据传输和其他应用程序的任务。

在这两小节中我们描述了 DVI 系统软件从最初的 DOS 环境下的 AVSS 到当前正在开发的下一代的系统软件 AVK 的发展过程。82750 硬件处理能力的增加以及希望把 DVI 移植到多种硬件平台和操作系统环境，这是 AVK 重要的设计目标。

Meyer 在他的一本叫做《面向对象软件结构》（《Object-Oriented Software Construction》）书中引进了一个“操作模型”的概念作为软件设计的起始点，他说“任何有用的系统必须基于某些工作现象一定的解释”。数字式生产演播模型已经证明在描述 DVI 系统特性以及一般的多功能媒体系统时，以上概念是非常有用的。它也为新系统潜在的用户讨论特性和技术要求提供特殊的语言。AVK 选择模型已经考虑了用理想一致的方法对增加新的特性和扩展能力给予足够的支持。

数据流和实体的分解是多媒体软件系统设计成功的关键，它允许软件扩充以便获得更先进的硬件系统的支持。从显示的位映像数据分离出解压缩位映像数据是一个例子。新的 AVK 系统已能演示由 I/O 带宽和处理器能力的扩充形成的多个数据流的功能。由于主 CPU 对微码引擎的编程，对视频特技效果增加专用的微码子程序，使视频效果得到翻新和扩展。通过改变下层软件结构能够执行新的改进的图像压缩和解压缩的算法。

最后，使用 82750 PB 微码实时地执行任务调度使主机 CPU 能够有更多的时间去完成文件的输入、输出以及其他众多的应用任务。任务调度的方法比起其他系统有很多优点：首先，它与主机 CPU 的结构以及操作系统的环境无关，AVK 几乎全用 C 语言写成，因此允许 AVK 保持更多的可移植性。未来任何一种 DVI 硬件结构为了实时调度视频任务都能够使用微码引擎。其次，这个技术的另一个特点是 82750 PB 难得空闲。在 AVSS/RTX 的情况下，A 级视频处理器（82750 PA）必须等待从主 CPU 传来的指令，等待主 CPU 调度下一个任务，因此，它有可能空闲。现在允许像素处理器调度它自己的任务，因此能够更充分利用 82750 PB。当主机 CPU 想改变微码操作时，注意不是在每一帧而是在需要的时候，主 CPU 才与 82750 PB 通信，传送微码子程序。因此，这种方法使 82750 PB 变忙，主 CPU 变闲，这正是我们希望

的目标。最后,这种方法不需要另外的处理器。微码引擎使用了 25MHz 的 82750 PB7%的时间周期。

AVK 进一步的研究方向是使其能在更多的操作系统环境下运行,如 UNIX/X Window 以及 Apple 的系统 7;改变使用 82750 PB/DB 硬件的结构,也就是说能够使 AVK 在其他硬件支撑环境下正常工作。

## 5.3 将多媒体和通信功能集成到 CPU 芯片中

计算机产业的发展趋势应该是把多媒体和通信技术融合到 CPU 芯片中,过去计算机结构设计较多地考虑计算功能,今天随着多媒体技术、计算机技术、网络技术的发展,计算机结构设计需要考虑增加多媒体和通信功能。

计算机最早是在大学和研究机构用于数学运算;三十年前计算机开始进入企业界,当时只被当作数值处理机,近一二十年计算机从机房搬到办公桌以后,使计算机增加了字符、文字、图形和图表的处理能力;最近几年由于多媒体和通信技术的飞跃发展,需要计算机具有综合处理声、文、图信息及通信等功能、使其能够广泛地用在多媒体信息管理系统、多媒体信息点播系统、视频会议系统、电子图书馆、交互式电视系统、远程医疗诊断系统,虚拟教室——远程学习系统以及计算机支持的协同工作系统等。

为了使现有的计算机较好地满足上述系统的各种需要,希望计算机在原有的硬件和软件支撑平台上增加如下功能:

- 多媒体数据的获取。
- 多媒体数据的压缩和解压缩。
- 多媒体数据的实时处理和特技。
- 多媒体数据的输出和多媒体通信。

### 5.3.1 集成的设计原则

在这里集成的含义是指:在原有计算机体系结构中,如何增加上述新的功能。它们的设计原则包括以下内容。

#### 1. 采用国际标准的设计原则

标准化是产业活动成功的前提,为了使新型的计算机增加多媒体数据的获取,压缩和解压缩、实时处理和特技、输出和通信等功能,设计时必须采用国际标准。前几年世界上许多国际性的大公司已经研制了很多种多媒体计算机系统,其中卓有成效的公司和系统有:Commodore 公司的 Amiga 系统、Philips/Sony 公司的 CD-I 系统、Apple 公司的 Quick Time 以及 Intel/IBM 公司的 DVI 系统,在这些系统中都有自己的视频音频信息压缩编码和译码算法,如 DVI 系统中的 PLV 算法。为了提高速度,提高产品效益,进行实时处理,这些公司还投资了大量研究经费,设计制造了专用的硬件芯片。如 Intel 公司的 DVIAction Media 750 II 系统专门设计了 82750PB 像素处理器,82750DB 显示处理器,82750LV.VRAM/SCSI/Capture 接口门阵,82750LA 音频和 Color Key 门阵以及 82750LH 主机接口门阵,在当时的国际市场上取得了良好的经济效益和社会信誉,该公司的 DVI 系统荣获了 1991 年秋季 Comdex 国际计算机博览会的最佳多媒体产品奖(Best Multimedia Product)及最佳展示奖(Best of

Show)。但是随着最近几年国际标准化组织 ISO 先后制定了 ISO/IEC 10918 连续色调静态图像的数字压缩和编码 (JPEG), ISO/ 11172 具有 1.5Mbit/s 数据传输率的数字压缩媒体运动图像及其伴音的编码 (MPEG-I), ISO/IEC 13818 带有伴音的运动图像的通用编码 (MPEG-II), ITU-T 协议 H.320 窄带 ISDN 可视电话系统和终端设备、ITU-T 协议 H.324 低比特率多媒体通信终端以及 ITU-T 制定的 H.261P×64kb/s 可视电话业务的视频编码等国际标准,使得上述产品的进一步发展和销售遇到了较大困难,必须改变设计采用上述国际标准,走标准化的道路。

## 2. 多媒体和通信功能的单独解决变成集中解决

计算机综合处理声、文、图信息和通信功能,过去的解决办法是设计专用适配卡分散单独解决,例如使用类似声卡解决声音的输入输出和实时编码、解码及处理问题;使用类似视频卡解决视频信号的输入、显示输出及多窗口的彩色键连问题;使用视频信号压缩编码和译码卡解决视频信号压缩和解压缩问题;使用局域网和 ISDN 网接口解决局域网和远程网的通信问题。现在希望采用微码引擎,设计制造合适的 DSP 或阵列处理机通过微码编程综合解决上述问题。几年前 TI 公司设计制造的 Mwave 系统就是一个很好的示例,Mwave 的多媒体特性如图 5-34 所示,它形象生动地说明了 TI 公司把多媒体和通信功能从单独解决变成集中解决的方案。过去要想使计算机具有 Fax 和 Modem 的功能,必须选用 Rockwell 公司的专用芯片,并设计一块专用的接口板;要想使计算机具有语音识别和语音合成的功能,可以选用 TI 公司的语音信号处理的专用芯片,并设计一块专用的接口板;要想使计算机具有图像处理、压缩编码和译码的功能,可以选用 C-cube 公司生产的图像处理、压缩和解压缩的专用芯片,并设计一块专用的接口板;要想使计算机具有音乐合成的功能 (MIDI),可以选用 Yamaha 的音乐合成的专用芯片并设计一块专用的接口板。现在可以选用 TI 公司生产的 TMS320 多媒体数字信号处理芯片,它采用微码引擎综合集中解决上述问题,通过加载不同的微码可以使 TMS320 Mwave DSP 分时具有 Fax、Modem、语音信号处理、图像信号处理及音乐合成的功能。

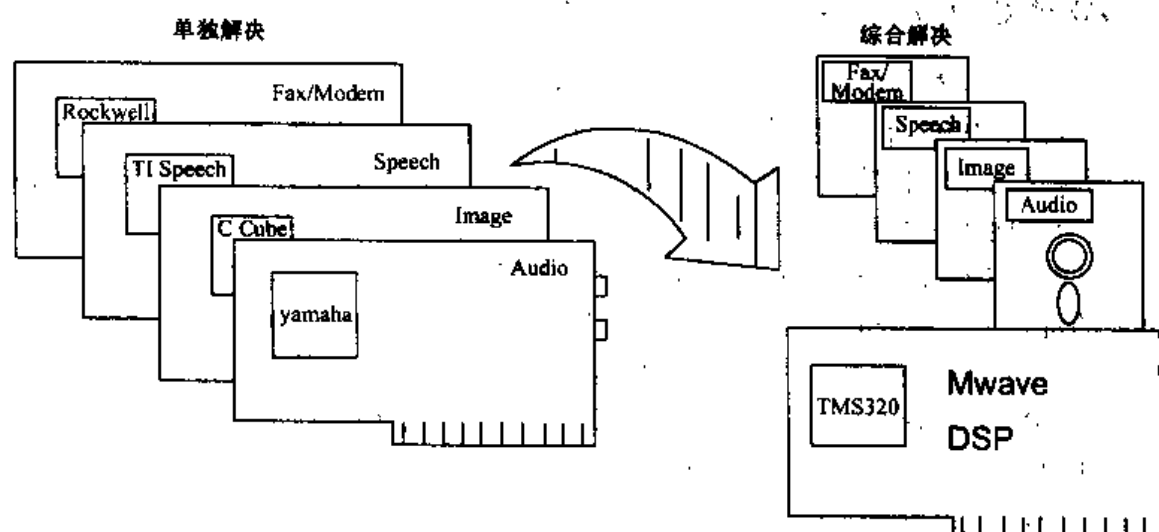


图 5-34 Mwave 的多媒体特性

### 3. 体系结构设计和算法相结合

要想使计算器具有综合处理声、文、图信息和通信能力的最佳解决办法是把计算机体系结构设计和算法相结合。综合处理声、文、图信息和通信功能算法的核心是数字信号处理, 数组向量运算, 即以乘加运算为核心的矩阵运算, 例如, 我们要执行符合国际标准的 JPEG、H.320、MPEG 等图像编码和译码以及通信的算法时, 要进行离散余弦变换 (DCT), Huffman 运算, 行程编码 (RL) 及变长编码 (VLD) 运算等。这类运算大量重复使用乘累加运算, 它们要求处理器具有如下的特点:

- 快速灵活的算术运算能力。
- 扩展的动态范围。
- 多操作数的同周期提取。
- 硬件的循环缓冲。
- 多个二维查找表 (TD-LUT)。
- 无额外开销的循环和分支控制。

### 4. 把多媒体和通信技术作到 CPU 芯片中

为了使计算器具有多媒体和通信功能, 最早的解决办法是采用专用芯片设计制造专用的适配卡; 其次是把多媒体和通信功能作到母板上, 最佳的方案是将多媒体和通信功能融合到 CPU 芯片中。从目前的发展趋势看可以把融合方案分成两类: 一类是以多媒体和通信功能为主, 融合 CPU 芯片原有的计算功能, 它们的设计目标是用在多媒体专用设备、家电及宽带通信设备上, 可以取代这些设备中的 CPU 及大量的 ASIC 及其他芯片, 它们的代表产品是 Philips 公司的 Trimedia, MicroUnity 的 Media Processor 及 Mpact Media Engine。另一类是以通用 CPU 计算功能为主, 融合多媒体和通信功能, 它们的设计目标与现有计算机系列兼容, 融合多媒体和通信的功能, 主要用在多媒体计算机中。它们的代表产品是 Sun 公司的 Ultra SPARC- I 和 II, Cyrix Multimedia 586、HP 公司的 MAX-2, Intel 公司的 MMX 及 Motorola 公司的 VeComp701。

### 5.3.2 多媒体处理器——Mpact 和 Trimedia

每年一次在美国举行的微处理器论坛 (Micro-Processor Forum) 将宣告即将来临的微处理器的流行样式。1995 年主要围绕着下一代 CPU 的问题, 而 1996 年论坛的热点论题转到多媒体而优化的专门的微处理器上, 它表明 1996 年流行的将是对多媒体用途的微处理器。为了更有效地实时处理模拟音响、视像的数字化数据流, 这些多媒体处理器都采用了数字信号处理器 DSP (Digital Signal Processor) 中采用的一些技术。实际上它们通常很像 CPU/DSP 的混合物, 更好的方法将是 CISC、RISC 和 DSP、Array Processor 结构的融合。随着 PC 从处理简单的数字和文字走向处理更复杂的数据类型, 旧的处理方法已不完全适用, 今后新一代的 PC 处理音响和视像将和今天 PC 处理文字和表格一样平常。

1996 年世界上很多厂商推出了多媒体处理器 (Multimedia Processors), 其中佼佼者 Chromatic Research 公司的 Mpact, Philips 公司的 Trimedia, MicroUnity 的 Media Processor (由 MediaProcessor, Media Codec 及 MediaBridge 三件组成) 以及 Nvidia NVI 高度集成的多媒体加速器, 它们引起世界舆论界普遍的关注, 人们认为它将会成为多媒体时代多媒体计算机的协处理器。Microsoft 公司总裁 Bill Gates 曾预测, 未来的个人计算机只有三个主要组



件，即存储模块、CPU 及一个执行复杂 I/O 的处理芯片，多媒体处理器便是一个可独立执行 I/O 处理的高速专用芯片。

1996 年 2 月位于美国桑尼维尔的 ChromaticResearch 公司开发的 Mpact 媒体处理器一经问世，就立即引起轰动，受到各种宣传媒体的普通关注：华尔街邮报、今日美国、Byte 杂志等数十家都刊登了一系列的评论。有人认为：“媒体处理器——多媒体时代的计算机芯片”。除了 Chromatic 公司，还有 IBM、MicroUnity、Mitsubishi、NEC、Philips、Samsung 和 SGS-Thomson 等众多有影响的大公司正在推出自己的媒体处理器。在下面两小节将重点介绍 Mpact 和 Trimedia 芯片、板卡及其软件。

### 1. Mpact 多媒体处理器

美国 Chromatic Research 公司是一家不生产半导体芯片的芯片设计和软件开发商，它采用了一种全新的商业模式，它负责 Mpact 多媒体处理器芯片的设计及软件开发，而芯片和评价卡的生产交给合作伙伴（东芝、LG、SGS-Thomson 等半导体厂商）去做，这样 Chromatic Research 公司就可以充分利用半导体厂家的现有资源，而不要在生产设备上投入巨额资金，从而将资金用于设计和研究中。

由 Chromatic Research 公司设计，LG Semicon、Toshiba 公司设计制造的 Mpact 芯片现有三种型号：

M1: Mpact/3000; M1 R75MHz; Mpact R/3600; M2: Mpact 2/6000

其中 3000、3600、6000 描述了芯片的速度，速度单位是 MOPS-Millions of Operation Per Second, M1 在 LG 公司的产品型号为 GM90DC701Q, 在 Toshiba 公司的型号为：TC 80301F。

(1) Mpact 芯片的功能。Mpact 芯片的主要功能如图 5-35 所示。这是 Mpact 芯片目前设计的已形成的七种多媒体功能，同时芯片设计采用了微码引擎（Microcode Engine），如果想增加新的功能，可采用软件升级的方式而不必更换硬件，这样可以有效地保护用户投资。具体功能如下：

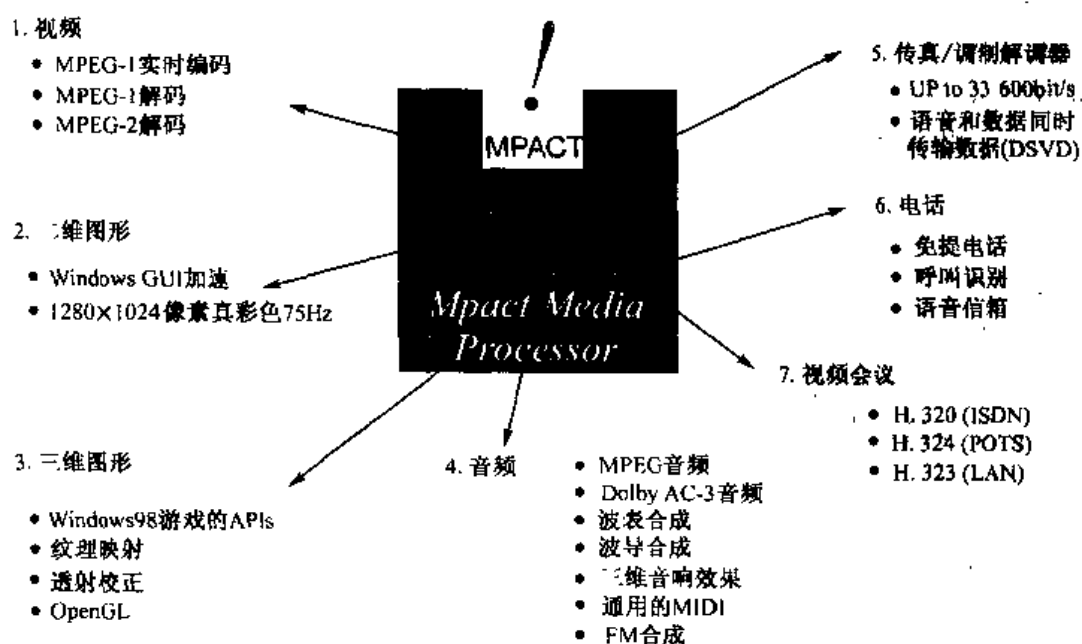


图 5-35 Mpact 媒体处理器功能图

1) 视频:

- MPEG-I 实时编码、30 帧/秒, 家庭视频编辑, 数字 VCR (视频捕捉+MPEG 格式回放)。
- MPEG-I 译码 (全屏幕、30 帧/秒、18 位彩色), VCD 播放、专利的动态影像处理技术达到最佳回放效果。
- MPEG-II 译码 (全屏幕、30 帧/秒)、DVD 播放。

2) 二维图像:

- Windows GUI 加速。
- SVGA、VGA 兼容。
- 1280×1024×18, 60Hz;  
1024×768×18, 60~85Hz;  
1024×768×24, 60Hz;  
800×600×24, 60~120Hz;  
640×480×24, 60~144Hz。
- 符合娱乐 PC 97 规范, 具有多制式电视功能。

3) 三维图形:

- Windows 95 应用程序接口。
- Z 缓存, 双缓存的动态显示; 折光、反射映照和阴影、线框图、实体模型。
- 透视纠正, 纹理映像。
- 支持点、线、三角形等图案元素。
- 400 万像素/秒, 80 万三角形/秒。

4) 音频:

- MPEG 音频。
- Dolby AC-3 音频。
- 波表合成。
- 波导合成。
- 三维声音和效果。
- 通用 MIDI。
- FM 合成。
- 声霸卡兼容。

5) 传真/调制解调器:

- V.34bis 33.6K 数据传输, V.17 一、二两级传真。
- V.80 在异步接口上同步数据传输, V.14 异步同步转换, 支持 RS 232 接口。

6) 电话:

- 免提式电话。
- 呼叫识别。
- 自动应答。
- 语音信箱。

7) 视频会议:

- H.320 (ISDN) 综合业务数字网。
- H.323 (LAN) 局域网。
- H.324 (POTS) 普通电话系统。

(2) Mpact 多媒体处理器的结构原理。Mpact 核心芯片使用  $0.5\mu\text{m}$  三层 CMOS 工艺, 由约 1.5M 个晶体管组成。Mpact 芯片工作电压为 3.3V, 它也能在 5.0V 电压下正常工作。Mpact 芯片工作频率为 62.5MHz, 芯片为 240 管脚的 FQFQ 封装芯片。

Mpact 芯片能处理各项多媒体功能, 其内部的 CPU 指令形式为“超长指令字格式”VLIW (Very Long Instruction Word)。指令的执行方式属于“单指令多数据”SIMD (Single Instruction Multiple Data) 方式, “单指令多数据”的指令执行逻辑更加适合于多媒体功能的处理。这是因为多媒体功能处理运算的特点是: 运算量大, 运算种类单一, 精度要求低, 速度要求高。SIMD 技术正适合多媒体功能的运算, 指令的处理在 Mpact 芯片内还采用了流水线 (Pipeline) 技术。在流水线中, 指令的执行分成四步: 取指、译码、执行、回写。流水线技术的使用也提高了 Mpact 的运算能力。

Mpact 芯片内部结构如图 5-36 所示。它是五个接口控制器, 一个多信道 SRAM 和 CPU 组成。五个接口控制器为存储器接口、PCI 总线接口、视频接口、显示接口及外设 I/O 接口。五个接口与 SRAM 通过总线和 FIFO 相连。多信道的 SRAM 作为主 CPU 的存储器, 其功能类似于 Cache。4KB 的 SRAM 有 8 个读写信道, SRAM 通过 8 个读写信道可同时支持各接口和 CPU 之间的数据交换。另外 Mpact 芯片中还有 792 位的称为 Crossbar 的内部连通线, Crossbar 连通芯片内的各功能部件, 使芯片内数据传输率可达 9.0Gbit/s。

1) Mpact 的 CPU 及指令处理单元。Mpact 的 CPU 由四个 ALU (Arithmetic/Logic Unit) 组和一个运动估计 (Motion Estimation) 单元以及指令译码控制器组成。

Mpact 芯片最小处理单元为 72 位。Mpact 内每字节由 9 位组成, 72 位为 8 字节。在 Mpact 中, 称 8 字节为双字长。每一个双字长内可有一或两条指令。一般的指令为 3~5 字节长。对于可执行向量运算和块移动操作的指令来说, 一条指令可以使 4 个 ALU 同时工作, 使 Mpact 芯片的最高运算处理能力达到 3.0BOPS (Billions of Operation Per Second)。

4 个 ALU 组的功能如下:

- ALU1: ALU1 是 Shift/align ALU 组。
- ALU2: ALU2 是通用 ALU 组。
- ALU3&4: ALU3 和 ALU4 联合用于处理多媒体的一些特殊运算。

运动估计单元由大约 400 个算术单元组成, 在 MPEG-I 或 MPEG-II 的运动估计运算中提供了 20BOPS 运算能力。

Mpact 芯片的指令有三字, 四字, 五字指令。典型的四字指令为一字操作码, 二字源操作字, 一字目标操作字, 而三字指令为一字操作码, 一字源操作字和一字目标操作字。五字指令提供了四个操作字, 这样允许执行向量运算。

Mpact 在执行时, 一般由两个指令组成一个“指令对” (Instruction Pair)。“指令对”长度为一个双字长, 除非指令之间有资源冲突, 两个指令可以同时执行, 一旦冲突发生, 则两条指令均单独执行。

2) 多信道静态随机存储器。4KB 的 SRAM 被划分为 512 个双字长, 执行软件将 SRAM 划分为“指令段”和“数据段”, “指令段”可以为 256, 512 或 1K 大小, “指令段”内

的指令顺序是由地址顺序决定的。“数据段”内地址空间则是自处理的。由于 Mpact 不提供保护模式，因而划分和管理“指令段”和“数据段”的工作由程序员完成。

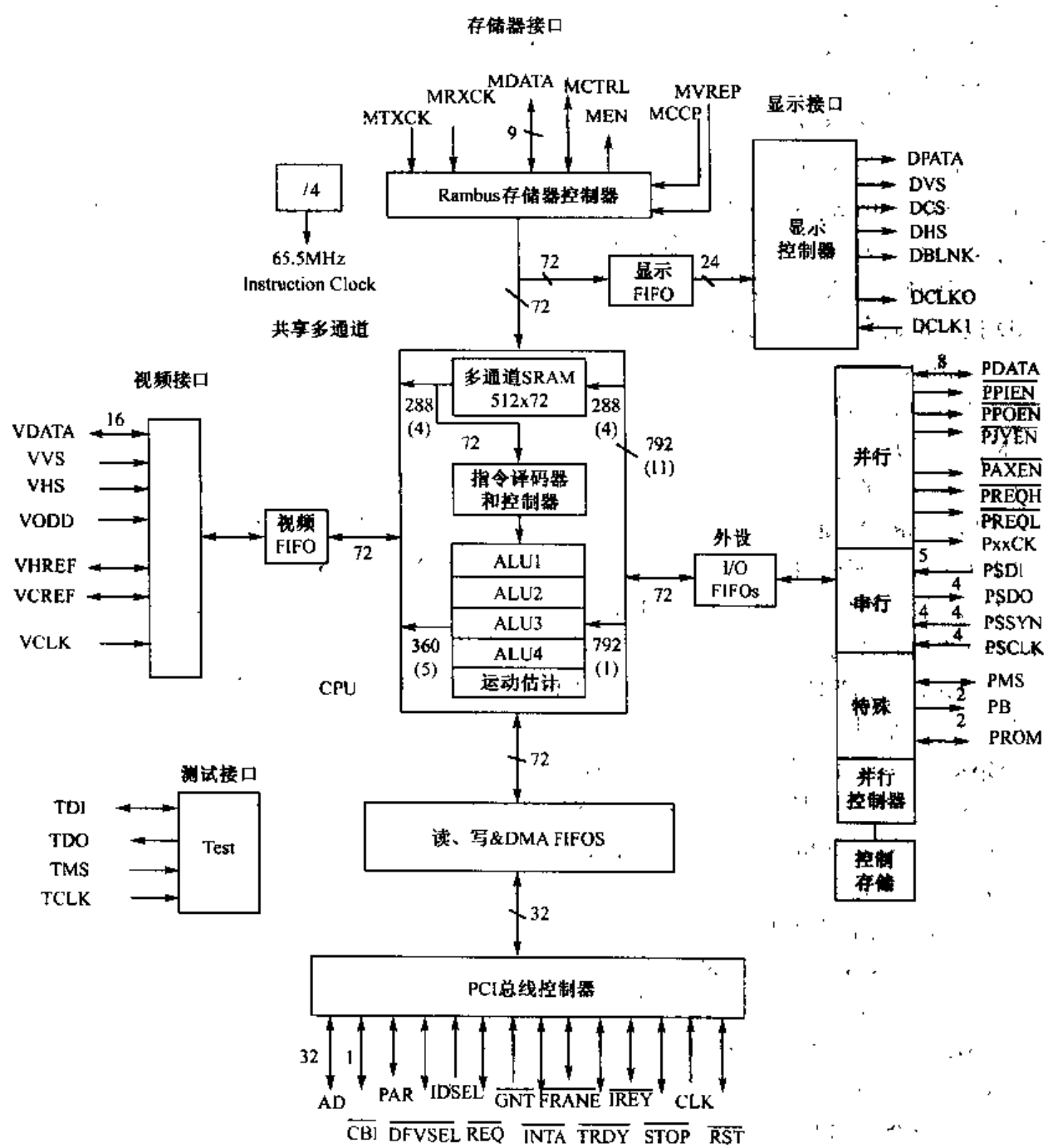


图 5-36 Mpact 结构原理图

SRAM 上有 8 个信道，其中 4 个为写信道，另外 4 个为读信道，这 8 个信道均可以同时操作。一个写信道被 RDRAM 专用，另一个写信道为 DMA 专用，另外两个写信道则为通用信道。

3) Mpact 的 RDRAM 接口。Mpact 卡上有一组 RDRAM 芯片作为板卡存储器。RDRAM 能够提供高达 500Mbit/s 的数据传输率，远远超过其他 DRAM 的数据传输率，Mpact 卡的 RDRAM 组的大小可在 2M~8MB 之间自由选择。

RDRAM 的控制器 RAC (Rambus Access Controllor) 位于 Mpact 芯片的内部。RAC 提供了对 RDRAM 的管理, 控制器能够直接读写 SRAM 和显示 FIFO, 地址转换也在 RAC 内部完成。另外, RAC 还提供了存储器刷新和 RDRAM 的初始化设置。

Mpact 的最小 RDRAM 要求为 2MB, 最大可安装至 8MB, RDRAM 的大小设置与显示分辨率的关系如表 5-2 所示。

表 5-2 RDRAM 容量与显示分辨率的关系

显示分辨率 (像素)	bits/像素颜色	最小的容量
1600×1200	8	4MB
1360×1024	8	4 MB
1280×1024	18	4 MB
1280×1024	8	2 MB
1152×870	18	4 MB
1152×870	8	2 MB
1024×768	24	4 MB
1024×768	8, 18	2 MB
800×600	8, 18, 24	2 MB
640×480	8, 18, 24	2 MB

4) PCI 总线接口。PCI Controllor 与其三个 FIFO 部件控制了 Mpact 芯片和主机 X86 的交互处理, 这种 PCI 的接口与 PCI 局部总线技术规范 2.0 版本完全兼容, PCI 接口支持 3.3V 或 5V 的电源信号, 32 位的 PCI 峰值传输速率为 132Mbit/s。

Mpact 芯片通过 PCI 接口可与 X86 或其子系统以 33MHz 的同步频率进行主或从的操作。当进行 DMA 操作时, Mpact 芯片作为主, 而当对 Mpact 卡进行读写操作时, Mpact 芯片作为从。

PCI 接口的三个 FIFO 分别为 DMA FIFO、读 FIFO、写 FIFO。DMA FIFO 长为 8 字, 而另两个 FIFO 长为 1 字, 如图 5-36 所示。

从主机角度讲, PCI 在地址空间占据三个部分: 128M 的存储器, 128KB 的扩展 ROM 和特殊设备的地址。图 5-37 为 PCI 的地址空间分配图, PCI 系统结构空间为 PCI 的设置空间。这段空间内有 PCI 的 ROM 空间和存储器空间基址, 指示了 ROM 和存储器空间的位置。传统硬件寄存器和 BIOS 为 ROM 和基于 PCI 板的模拟显示卡和声卡的功能寄存器, PCI 的 128KB ROM 空间指的就是这个区域。PCI 地址区域的顶部为 128MB 的 RAM 地址空间。这里地址空间分配给帧缓存、多信道 SRAM、扩展的 ROM 和内部寄存器。在实际情况中, 128KB 的 ROM 空间和 128MB 的 RAM 空间均有很大的空闲。如 128MB 的 RAM 空间, 最多只能占用 10MB 左右。

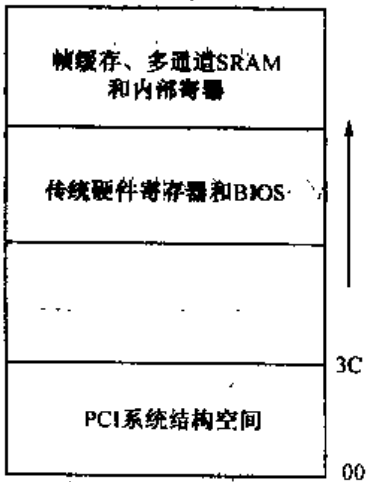


图 5-37 PCI 的地址空间分配图

5) Mpact 视频接口。视频控制器与 32 字节的视频 FIFO 相连, 在 16 位的视频接口上传

送 YUV 格式的视频数据。视频接口是可编程的双向接口，但是传输只能工作在半双工的状态下。工业标准的 NTSC 和 PAL 的编码器、译码器均可在全速率扫描模式下提供模拟信号或 S-Video 信号。

6) Mpact 的显示接口。显示控制器与 512 字节的显示 FIFO 相连，它为显示提供连续的 RGB 信号。显示控制器控制全部的数据，并产生同步信号和时钟信号，提供给 RAMDAC 或显示编码。显示控制器支持 8 位的索引色 (Index Color) 或 24 位的真彩色 (True Color) 模式，也支持 NTSC 或 PAL 的 RGB 编码器。

RAMDAC 的信号是 Mpact Serial 总线或外设 I/O 总线 (Peripheral I/O Interface) 提供的。

显示数据可以用一字、双字或三字格式存储，以支持 8 位、15 位、16 位、18 位或 24 位。Mpact 的特殊格式 Chro-Color 为 18 位，正好占用两个字节。

显示数据从记忆体格式到 RAMDAC 格式的转换是在显示控制器内完成的。  
显示的功能逻辑由显示 FIFO，显示前端 (Front End)，显示后端 (back End) 和时钟发生器完成，如图 5-38 所示。

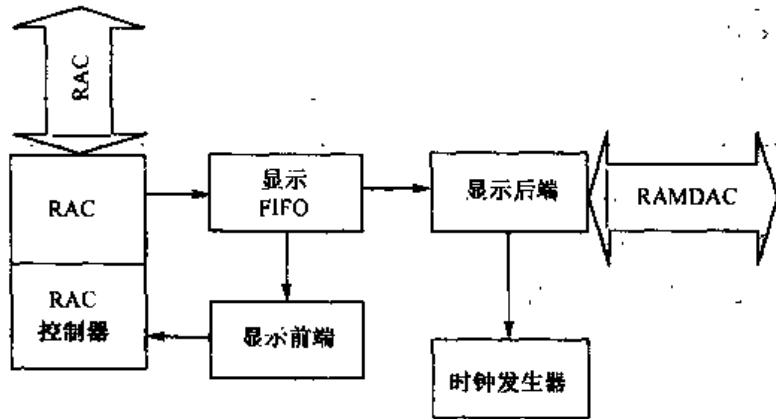


图 5-38 Mpact 显示接口原理图

显示 FIFO 将数据由 RDRAM 帧缓冲区移至显示后端。  
显示前端为 RAC 提供帧存的地址，光标的地址及格式。  
显示后端将 RGB 信息送入 RAMDAC 并提供 Zoom 控制。  
时钟发生器提供同步时钟脉冲。  
Mpact 支持的分辨率如表 5-3 所示：

表 5-3 Mpact 的外设 I/O 接口

显示分辨率 (像素)	像素时钟频率 (MHz)	最大刷新速率 (Hz)	最大的像素彩色位数
1600x1200	220	75	8
1360x1024	135	75	8
1280x1024	125	75	18
1152x870	100	85	18
1024x768	80	85	18

(续)

显示分辨率 (像素)	像素时钟频率 (MHz)	最大刷新速率 (Hz)	最大的像素彩色位数
800×600	50	85	24
640×480	32	85	24
NTSC: 720×480	12.27	30	24
PAL: 720×576	14.75	25	24

7) Mpact 的外设 I/O 接口。Mpact 的外设 I/O 接口 (Peripheral I/O Interface) 可以分为三个部分: 一个 8 位双向并口, 4 个双向串口以及 3 个特殊功能总线。

Mpact 的外设 I/O 接口提供最高传输率为 5Mbit/s 的数据吞吐能力。

Mpact 的串口共 4 个, 每个串口由 PSD[i]、PSDO[i]、PSCLK[i] 及 PSSYN[i] 4 根控制线, 数据线 ( $i = 0 \sim 3$ ), 4 个串口中的口 C 连接 ROM, 口 C 的硬件编码设计支持 24 位的地址寻址, 读取 ROM 内的数据。口 A、B、D 为通用口。

Mpact 的并口为 8 位宽, 19 位信号/数据线, 并口连接了外设 I/O 接口的控制逻辑, 用以扩展并口功能。在 Mpact 的设计中, 并口通过与 Mpact 系统集成 ASIC (MSIA) 相连, 将 8 位数据并口线扩展为 80 位的信号、数据并口线。

80 位的信号、数据并口线为 24 位地址线、24 位输入信号线、16 位输出信号线、8 位数据 I/O 线以及 8 位的 Joystick/MIDI 信号输入线。

24 位的输入信号线又被分为两个部分: 12 位为中断请求信号线, 另外 12 位为 DMA 请求信号线。

16 位的输出信号线也被分为两个部分: 8 位为片选信号线, 另外 8 位为通用输出控制。外设 I/O 并行总线的功能可归纳为两点:

(a) 为并口外设提供高速的传输, 如 Audio Codec、Telephone、MIDI VART 等。

(b) 控制 Mpact 视频、Mpact 显示等有设置和控制参数的设备的特殊接口, 如:

- Mpact 的串行总线: 2 位, MSB1 与 MSB2。双向数据传输, 控制信号传输位。
- 可编程位 (PB): 2 位, 单向输出, 用于与外设进行通信, 由外设的可编程控制器来设置和清除。
- 系统标识 ROM 位: 1 位, 双向数据传输, 它与 Mpact 卡上的设备标识 PROM 相连, 用于写入或读出每块板卡惟一的标识码。

## 2. Trimedia 芯片、参考板及其软件开发环境

TM-1000 是 Philips 公司于 1996 年底推出的新一代媒体处理芯片 (Media Processor)。它是一款针对实时处理音频、视频、图像和通信数据流的通用微处理器, 内部集成了一个极高性能的 CPU, 一些周边的 I/O 单元和协处理单元。综合运用内嵌 DSP 的方案和通用 CPU 的高度可编程特性, 使它以极高的性价比实现了高性能的多媒体功能。因此它在消费类电子产品和 PC 产品中得以广泛应用。它有以下几个主要特点:

- 在一块芯片上同时处理音频、视频、图像和通信数据流。
- 内部集成一个强大、通用的非常长指令字 VLIW 的处理器内核, 包含分离的数据和指令 cache (高速缓冲存储器), 峰值计算速度达到 40 亿次/秒。
- 相互独立, DMA 驱动的多媒体输入输出单元接受和输出格式化的数据, 以及能实现

特殊多媒体算法的多媒体协处理单元。

- 指令集包括 RISC、多媒体、DSP 以及和 IEEE 兼容的浮点运算。
- TM-1000 内部和功能单元之间通过一个高性能的总线和存储系统进行通信。
- 利用高性能的软件开发工具和预先构造的库，使多媒体应用的开发基于 C 或 C++ 语言。

(1) Trimedia 处理器。参看图 5-39 TM-1000 的功能模块图，下面对 TM-1000 各个功能部分给以详细的介绍。

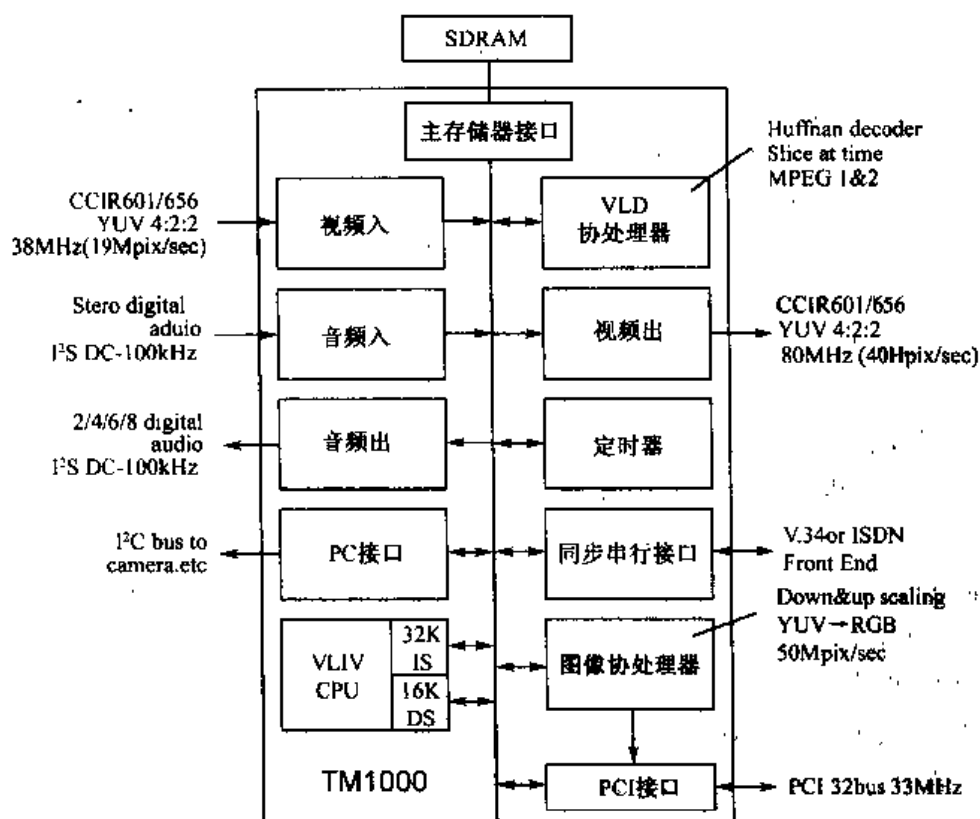


图 5-39 TM-1000 功能模块图

1) 可编程的 VLIW CPU。TM-1000 的心脏是一个强大的类似 DSP 的 32 位处理器核心。处理器实现了 32 位线性地址空间，内部包含了 27 个流水线驱动的功能单元和 128 个通用寄存器，任何寄存器都可以被任何操作、任何操作数利用。它的高度并行的 VLIW 体系结构实现了一个 five\_issue\_slot 引擎。核心使用 VLIW 指令集，每个时钟周期内，允许同时发出 5 个运算，这些运算可以针对 27 个功能单元的任何 5 个，其中包括整数单元、浮点数运算单元、类似于 DSP 的数据并行单元。大部分通常的运算一个时钟周期即可得到结果，复杂一点的运算需要多一点的时钟周期。

TM-1000 的指令集包括通常的 RISC（精简指令集）运算、实现强大 SIMD 功能和定制多媒体功能的专门 DSP 运算以及 32 位 IEEE 兼容的浮点运算，支持高字节顺序和低字节顺序。

TM-1000 的 CPU 还对指令和数据断点提供了特别的支持，这在应用程序的开发和调试



中尤其有用。

2) 专门的数据和指令 cache。TM-1000 内部具有专门的 16KB 数据 cache 和 32KB 指令 cache 支持 32 位处理器。为了提高 cache 的性能, 两种 cache 都使用了一种锁定机制, cache 的数据一致性由软件维持。数据 cache 是双端口的高速缓冲存储器, 它允许两个同时访问。为了减少内部总线的带宽需求, 存储器和 cache 里面的指令采用压缩格式, 在指令被 CPU 处理之前, 压缩的指令由指令 cache 解压缩单元完成解压缩。

3) 无缝的存储系统接口。TM-1000 存储系统通过使用片上 cache 和一个对外部 SDRAM (Synchronous DRAM 同步动态随机访问存储器) 无缝接口相结合的机制找到了性能和价格的平衡点。它的存储器接口有了足够的驱动能力, 可以直接驱动 100MHz, 8MB 的存储系统。利用较低的系统时钟频率和外部的缓冲器可以实现更大的存储空间。可编程的时钟能使存储系统和 CPU 可以运行在不同的时钟速度下。支持不同的存储器类型、总线宽度、时钟速度, 使 TM-1000 可灵活设计满足不同系统的需求。

4) 高速内部总线(数据高速通路)。TM-1000 内部的高速数据总线将内部各功能单元连接在一起, 并且提供对各功能单元的控制寄存器、外部的 SDRAM 以及外部的 PCI 总线的访问。它由分离的 32 位数据总线和地址总线组成: 总线传输协议采用块传输协议。片上的接口设备单元和协处理器单元可以是总线传输的主设备也可以是从设备。总线带宽的分配是可编程的。内部总线有一个中央判优器, 它有许多不同的工作模式, 允许针对不同的应用选用不同的判优算法。根据判优模式, 它控制对总线的访问和分配不同的带宽给请求单元。每种模式保证一个单元最小的带宽和最大的服务延迟。这种总线分配机制是使得 TM-1000 成为真正的音频和视频实时系统的特点之一。

5) 视频输入单元 (Video In)。任何 CCIR601/656 (8 位并行输出, 4:2:2 的 YUV 时间复合信号) 兼容的设备, 如数字摄像机, 都可以和视频输入单元直接连接。与 CCIR 不兼容的设备可以通过一个数字视频译码芯片 (如 Philips SAA7111) 与视频输入单元相连接。视频输入单元从片下单元获取数字视频, 将复合 YUV 数据分离, 如果需要的话可以进行子采样, 然后回写到 SDRAM 中。许多数字摄像机获取 640 像素每行或 720 像素每行的图像, 视频输入单元自身对图像进行水平子采样, 直接转换到 320 像素每行或 360 像素每行的图像, 转换过程不需要 CPU 的介入, 如果需要较低的分辨率, 那么获取信号过程中进行的子采样能够极大的减少存储和带宽需求。

6) 视频输出单元 (Video Out)。视频输出单元执行和输入单元相反的功能。它从 SDRAM 中分离的 YUV 数据结构产生 8 位复合的 YUV 数据流。它能执行任何可编程的任务, 然后将数字视频数据输出到片下 (off\_chip) 的视频子系统, 例如数字视频编码芯片、数字视频录相机或其他 CCIR656 兼容的设备。视频输出单元可以用 PAL 或 NTSC 制式输出连续的数字视频, 最高频率可以达到 40M 像素每秒。在产生复合视频数据流的同时, 视频输出单元可以进行可选的水平 2 倍放大, 将 CIF 的分辨率转换到 CCIR 的分辨率。为了能够同时显示图形和活动视频, 它也能在输出的图像的任意位置以任意尺寸产生复杂的图形覆盖 (如 alpha 混合)。

7) 音频输入和输出单元 (Audio In/Out)。TM-1000 集成了音频输入和输出单元。他们自动利用 DMA 方式为片下 (off\_chip) 的串行音频 AD 和 DA 提供数据流服务。两个单元都可以直接支持串行的 16 位立体声输入输出, 采样频率最高可以到 100KHz。配合少量的

外围电路, 可以支持 8 路声道的输出。音频接口是高度可编程的, 能够适应专门协议和未来的标准。TM-1000 可以编程, 提供主时钟给外部的 ADC 和 DAC。片上时钟发生器的精度可以达到 0.0006ppm。如此高的分辨率, 使编程者能精确地控制采样时钟频率, 以简化复杂多媒体系统中的同步算法。

8) 专用的协处理器。TM-1000 具有很强的处理能力, 又有便宜的价格, 关键之一是它拥有专用的硬件协处理器。

9) 图像协处理器 ICP (Image CoProcessor)。ICP 和 CPU 下载图像处理和调整任务, 例如将图像从 SDRAM 中复制到显示卡的帧缓存中。它也可以当做一个存储器到存储器设备或存储器到 PCI 总线的协处理设备。在存储器到存储器的模式下, 可以执行水平或垂直的图像滤波和任意图像放缩, 这是它能执行输入像素 5 邻域的 32FIR (Finite Impulse Response 有限冲击响应); 滤波, 滤波系数全是可编程的; 在存储器到 PCI 总线的模式下, 可以执行水平放缩, 然后执行色彩空间变换。例如, 如果一个像素数组将要显示在一个 PC 的图形用户接口的窗口中, ICP 首先在存储器到存储器模式下调整图像垂直大小 (如果需要的话), 然后在存储器到 PCI 总线模式下调整水平大小, 最后执行 YUV 到 RGB 的色彩空间变换。ICP 也为重叠窗口中的活动视频显示提供支持, 窗口的数目和大小仅仅受到带宽的限制。最后经重新采样和变换的图像像素通过 PCI 总线传送到片下的图形卡或帧缓存。

10) 变长译码器 VLD (Variable Length Decoder)。VLD 帮助 CPU 译码非常高位速率的哈夫曼编码的视频数据流, 如 MPEG-1 或 MPEG-2 数据流。VLD 工作在存储器到存储器的模式下。CPU 传递给 VLD 一个指向哈夫曼编码的比特流的指针, 结果产生一个为 MPEG 解压缩软件优化的符号化比特流, 因此 CPU 和 VLD 之间的通信是最小的。

11) I<sup>2</sup>C 接口。TM-1000 的 I<sup>2</sup>C 接口使片内设备连接、控制外部 I<sup>2</sup>C 设备成为可能。这样, TM-1000 可以配制和监测外部视频设备的状态, 如视频译码器和编码器以及某些数字摄像机。它也用来在启动的时候从 EPROM 中读取启动程序。

12) 同步串行接口 SSI (Synchronous Serial Interface)。TM-1000 的 SSI 接口为多种多媒体应用提供访问通路, 例如电视电话或视频会议和 PC 机一般的数据通信。SSI 包含了和模拟调制解调器相联接所需要的所有的缓冲区和逻辑 (电路)。当和 Trimedia 的 V.34 软件库组合起来时, SSI 提供了完全和 V.34 兼容的调制解调器能力。另外, TM-1000 的 SSI 也能和 ISDN 接口芯片相连, 以提供更好的数字调制解调器能力。

13) 定时器。TM-1000 包含了 4 个定时器, 程序员可以使用其中 3 个, 另外一个保留给系统。

14) 高速 PCI 总线接口。TM-1000 的 PCI 总线接口将 VLIW 的 CPU 和片上的输入输出单元以协处理器单元联结到 PCI 总线。在基于 PC 的应用中, TM-1000 可以直接联结到标准 PCI 总线上, 因此它可以直接安装到 PC 的母板或是插入卡上。在 TM-1000 作为主处理器的嵌入式应用中 (embedded application), PCI 总线可以用来连接其他一些接口设备, 这些接口设备实现了某些 TM-1000 未提供的功能。

(2) Trimedia 软件开发环境。按照传统的做法, 嵌入式的微处理器应用程序用汇编语言书写代码, 虽然代码执行效率较高, 但却损失了代码的移植性, 而且也使产品开发周期变

得很长。Trimedia 提供了一套复杂的系统软件工具帮助程序员使用标准的 C 或 C++ 语言开发多媒体应用程序。这一套工具包括编译器，调试器，优化工具和仿真器。Trimedia 的软件开发环境 SDE (Software Development Environment) 保证了快速开发新的代码和移植已有的代码，从而缩短新产品开发周期，迅速占领市场。在 Trimedia 系列芯片中，软件的兼容性只限于源代码 (C 或 C++) 级别，强大的编译器使得程序员没有必要费力去书写没有任何移植性的汇编源程序。SDE 集成了许多通用 CPU 和实时系统开发环境所具有的共同特征，它由以下一些模块组成。图 5-40 的框图说明了 SDE 各模块之间的关系。

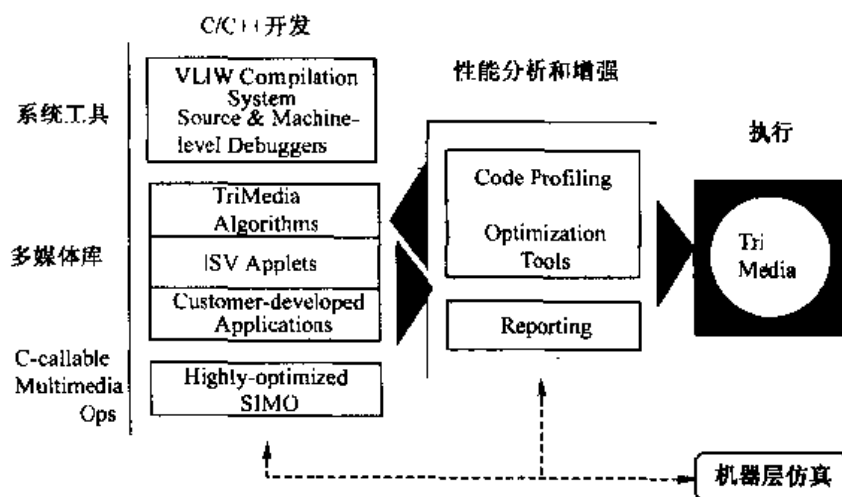


图 5-40 Trimedia 软件开发环境的组成模块

- VLIW 的标准 C 或 C++ 语言的编译器。
- 基于 GUI (图形用户接口) 的多级调试器。
- 性能分析和增强工具。
- 中间代码和机器代码级的仿真器。
- 一个实时操作系统 (RTOS) 内核。

1) VLIW 编译系统。图 5-41 说明了 Trimedia 编译系统对 C 或 C++ 源程序的操作过程。非常简单，Trimedia 的编译系统从 C 或 C++ 语言的源代码产生可执行程序。它模块化的设计使程序员可以完全控制编译、调试和优化的每一步。中间代码和机器代码级的仿真器允许程序员在开发过程中评估指令级的并行度和校验程序的逻辑功能。

在编译块完成 C 语言的预处理 (preprocessing) 或 C++ 语言的翻译后，核心的 C 编译器将 C 代码翻译成以判定树形式表示的中间代码。根据需要，可以使用中间代码仿真器来检测中间代码的并行度，这个过程和目标处理器的配置无关 (如 issue\_slot 的数目，功能单元的数目等等)。

指令调度模块 (scheduler) 检查判定树，并且产生针对目标处理器的 VLIW 指令。在这步处理过程中，调度模块给每条指令增加条件代码，这种技术能极大地减少代码的分支，因此也减少了程序执行时间。

在代码的编译和安排阶段，编译系统可以执行很多自动和程序员控制的优化操作。下面的性能分析和增强工具一节中将给以详细介绍。

当获得较好的并行代码后，代码和一个或多个目标模块或目标库组装、连接起来，然后

指令压缩模块压缩代码以增加代码的密度，这样就缩短了指令 cache 和片下存储器之间传输时间。这时，可执行程序就可以被装入存储器，并且在处理器或仿真器上运行。

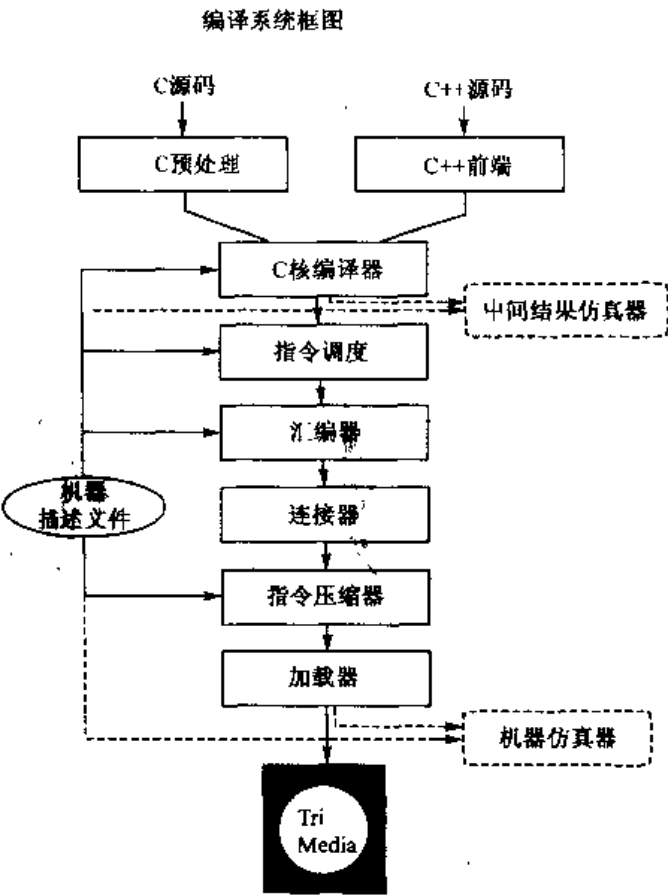


图 5-41 Trimedia 的编译系统

通过读入不同的机器描述文件 MDF (Machine Description file)，编译系统模块可以为任何的 Trimedia 体系结构产生可执行代码，不论是现在的，将来的或是正在实验的体系结构。MDF 描述了特定的 Trimedia 处理器使用的参数，如 issue\_slot 的数目，内部功能单元的类型和数目，完全的指令集等等。

Trimedia 的编译提供了用标准的 ANSI C 语言进行系统级编程的能力，包括对中断处理例程和对数据 cache 精细控制的支持。另外，尽管绝大多数的汇编级指令可以在源代码里实现，Trimedia 还是提供了完整的汇编语言。

2) 基于 GUI 的多级调试器。Trimedia 软件开发环境支持应用程序中间代码和机器代码级的调试。交互的、基于 GUI 的源代码级调试器使应用程序开发者能完全控制程序的动态执行，它使用编译器产生的符号化调试信息，提供检测终止程序的状态、观察变量和表达式的赋值、在代码里设置断点等功能。机器代码级调试器提供了相同的功能，使程序员可以每次单步执行一条 VLIW 的指令。

3) 性能分析和增强工具。开发高度并行的应用程序代码是一个需要复杂工具和程序员能力的交互的过程。运行在 Trimedia 细颗粒 (fine-grain) 并行处理器上的特定的应用程序的性能依赖于编译器对程序指令级并行的开发程序。为了帮助多媒体开发者获得高度并行和高性能的程序，Trimedia 的编译系统集成了很多有用 (强大) 的编译选项和分析工具。

(a) 代码框架分析 (Code Profiling)。它为应用程序的执行情况提供可靠的统计数字, 这为增加代码的并行度提供了途径。Trimedia 的代码框架分析帮助程序员获得最大可能的并行度和代码执行程序。当代码框架分析有效时, 程序的仿真执行能产生判定树和执行概率的统计数字。然后, 利用产生的代码框架分析信息重编译该程序以期增加程序的并行度。这个过程可以对程序的部分模块和整体重复操作, 以精细调整程序的性能。

(b) 判定树嫁接 (Decision Tree Grafting)。通过减少分支数目, 判定树嫁接可以增加指令级的并行度, 提供每个时钟周期内更有效的操作码。在此过程里, 从判定树上跳转或退出被它的目标判定树的副本所代替。嫁接过程可以被编译器自动执行, 或通过读入一个包含控制参数的文件而被程序员引导执行。这些参数使程序员可以控制代码密度, 其中包括最小概率阈值, 最大代码复制因子, 最小执行计数阈值, 最大嫁接深度以及嫁接使能。

(c) 别名分析 (Alias Analysis)。编译器自动执行别名分析以决定是否可以采取某些优化措施, 这是提高代码并行度的一个关键方法。编译器检测两个存储器单元是否相同或相互重叠, 如果两个都不是, 编译器试着削弱记忆体操作的阶数 (强度), 这样允许更多的操作并行执行。编译器支持 3 个层次的别名分析。

(d) 受限指针 (Restricted Pointer)。当编译器自动执行不安全的别名分析时, 受限指针使程序员外部强制编译器取消对该指针的别名分析。编译器利用这些信息去执行其他的优化操作。

(e) 局部和全局优化 (Local and Global Optimization)。通过对源代码的假定级别, 编译器对局部判定树或全局判定树支持 4 个层次的优化。

(f) 关键路径 (Critical Path) 绘图工具。Trimedia 的编译系统工具可以以图形方式输出关键路径以识别程序的并行。图形数据以 Post Script 格式输出。

(g) 基于 GUI 的 cache 分析器。通过优化使用 cache 可以获得最高的性能。SDE 包含以图形方式显示动态 cache 使用统计数字的工具。分析指令和数据 cache 命中失败可以帮助程序员找到调整代码和数据的时机, 以增加并行度。

除了上面提到的优化方法, Trimedia 编译系统还支持增值复制、常数合拢 (folding)、死代码消除、循环重复优化 (loop iteration optimization) 和局部公共子表达式消除。

4) Trimedia C-调用 (C-callable) 的特殊运算。这些高度并行的定制 (custom) 运算大大加速了数字信号处理和多媒体应用中常见的特殊运算的性能。定制运算提供了类似 C 函数调用的方式, 源代码里的访问语法和 C 语言保持一致。这些运算被自动安排 (调度), 充分利用 Trimedia 高度并行的 VLIW 体系结构, 可以提高应用程序的性能, 又减少了代码开发时间。

5) Trimedia 的应用链接库。通过提供多种处理音频、视频、图形和通信的多媒体算法, Trimedia 的应用链接库缩短了很多标准的多媒体应用程序的开发周期。这些库包含了 C 语言调用的例程, 例如 MPEG-I 和 MPEG-II 的译码, V.34 调制解调器, 浏览器, H.32X 视频会议, 音频合成, 2D/3D 几何造型等等。

6) 中间代码和机器代码级的仿真。两个代码级的仿真使程序员在程序开发过程中就有机会评估代码的性能和并行度。在代码的初期开发阶段, 中间代码可以进行仿真, 允许交互的调试和估计指令级的并行度, 测试程序的性能。中间代码级仿真假设 CPU 有无数个通用寄存器和功能单元。在代码开发的后期, 机器代码级的仿真器使程序员能够测试和提高应用

程序的性能。它能用来在一个实时操作系统里给定数目的时钟周期内单步执行单个、复位位的模块或者仿真整个程序。机器代码级的仿真器精确地模拟 Trimedia CPU、存储子系统、系统接口设备单元、中断处理、时序和 PCI 接口。通过读入 MDF 文件, 机器代码仿真器能够测可执行程序在不同的 Trimedia 体系结构上的性能。

7) 共享库、动态库的支持。Trimedia 的编译系统还包括一个可重入的 ANSI C 库, 以及其他共享库和动态库, 连接和装入这些库, 有利于减少可执行代码的大小。

8) 实时操作系统 (RTOS) 内核。多媒体应用要求系统资源和活动得到高效的管理, Trimedia 处理器支持 pSOS+ 实时多任务操作系统内核。pSOS+ 操作系统由 Integrate Systems Inc (ISI) 开发, 它基于开放操作系统标准 (Open System Standard), 并且针对多媒体应用作了专门的优化。它具有以下特点:

(a) 真正实现抢先式, 基于优先级的任务调度。pSOS+ 内核调度、管理和分配系统资源 (如 CPU、定时器、存储器、输入输出设备以及总线带宽等)、协调系统内异步的活动。pSOS+ 内核简单地将应用程序看成一个任务、输入输出设备驱动程序和中断服务例程 (Interrupt Service Routine) 的组合。一个任务是系统内竞争它自己的系统资源的最小单位。核心响应系统调用发送给它的信号, 在不同的任务之间进行切换, 从而使相互独立的任务得以并发执行。

(b) 合理的中断处理。ISR 处理中断并执行所要求的动作, 如重新设定设备, 读/写数据等。ISR 也驱动一个或多个任务去响应和处理与中断相关的情况。为了更好地响应实时应用和异步特性, 内核实现了真正的抢先式和基于优先级的任务调度。内核保证任何时刻正在运行的任务在所有准备好可以运行的任务中具有最高的优先级。更高优先级的外部事件可以抢先当前的任务。通过使抢先特性有效或无效, 或改变调度任务时间片的大小, 用户可以改变调度的行为。pSOS+ 核心通过核心外部的 ISR 来处理中断。为了提供可能的最快响应时间, 中断可以直接被传递给 ISR。从 ISR 发出的系统调用返回到 ISR, 消除了核心调度的时间消耗。

(c) 动态的, 基于对象的多任务。pSOS+ 是一个动态的、基于对象的核心, 它充分利用了为任务到任务通信、ISR 到任务通信、同步和互斥服务的自由竞争同步机制的优点。它支持的对象包括任务, 存储区域, 固定和可变长的消息队列, 事件, 定时器, 信号量和异步信号等。

(d) 动态的存储分配。pSOS+ 核心存储管理提供动态分配可变尺寸的段和固定尺寸的缓冲区的能力。

(e) 灵活的计时管理和时钟服务。pSOS+ 核心提供许多有用的函数, 其中包括维持系统时间、日期, 超时设定任务, 唤醒任务, 周期性的外部事件的超时设定, 跟踪一个正在运行的任务的时间片等的函数。

(f) 先进的错误处理和故障恢复。当系统遇到致命的故障时, pSOS+ 核心使用一种特殊的机制依次挂起任务。

(3) Trimedia 设计参考板。下面通过 Philips 公司提供的一块参考设计板, 介绍利用 Trimedia 处理器进行设计的基本原理。我们也将看到, 由于 Trimedia 处理器结构化的设计, 我们只需增加一些外围的功能器件, 就能组成一套高性能的多媒体应用系统。

图 5-42 是 Trimedia 参考板框图。这块 PCI 多媒体板卡以 Trimedia 处理器为核心, 配合

以相应的输入输出芯片，组成了一个实时、高效的多媒体系统。它具有 NTSC/PAL 制视频输入端，与 CCIR601 兼容的视频输出，通常两路的立体声输入和输出，6 路 AC-3 中高品质立体声输出，调制解调器模块以及标准 PCI 接口；另外配合相应的软件，它还可以完成各种复杂的视频和音频的压缩和解压缩。

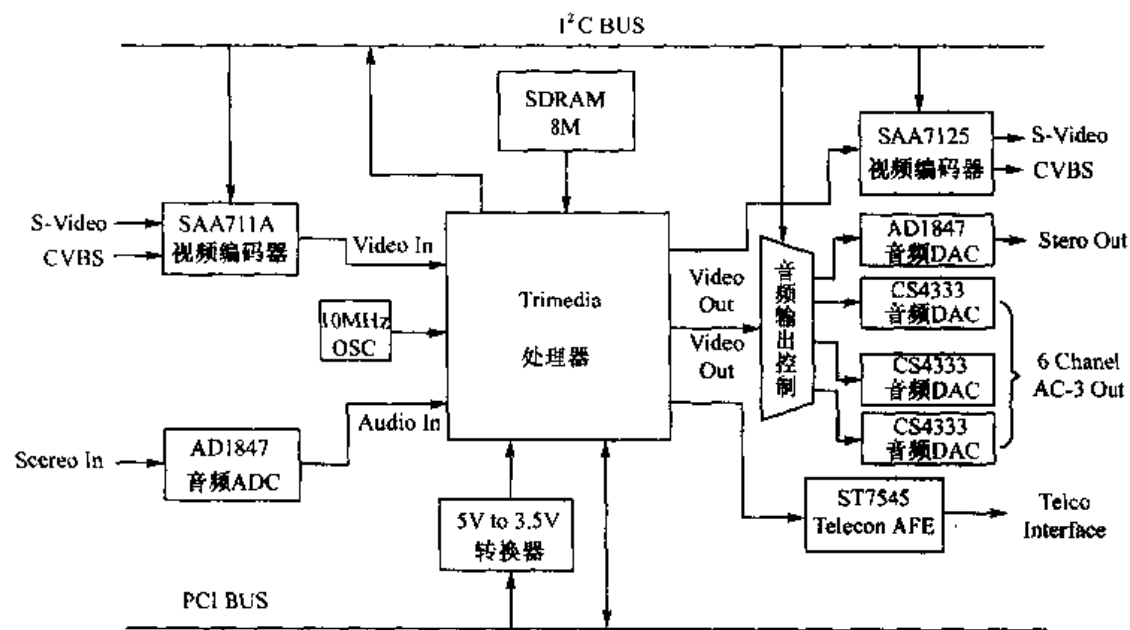


图 5-42 Trimedia 设计参考板与框图

视频信号输入处理和译码的核心是视频输入处理器 (Video In Processor VIP) SAA 711A。VIP 内部包含了两信道的模拟预处理电路 (包括信号源选择, 反走样滤波, A/D 转换器, 自动钳位增益控制), 时钟发生电路, 多制式数字译码器 (DDigital Multi Standard Decoder DMSD), 亮度-对比度-饱和度控制电路和一个色彩空间变换矩阵, 它最多可以处理 4 路信号源, 其中包括 NTSC/PAL 制的亮度和色度信号以及比家用录像机高一档次的 S-VIDEO (射频) 电视信号。由于 SAA711A 可以输出与 CCIR601 兼容的视频数据流, 因此它的输出直接与 Trimedia 处理 Video In 单元相连, 作为系统的视频处理前级, 向处理器直接提供与 CCIR 601/656 兼容的 YUV 时间复合信号, 实际上它也可编程输出数字化的 RGB 颜色分量。它还同时向处理器提供各种同步信号: 水平同步、垂直同步、水平参考、垂直参考信号等等。它还包含一个 I²C 接口, 通过这个接口它可以接受 Trimedia 处理器的控制信号, 设置和调整自己的工作状态。

视频信号输出处理和编码的核心是数字视频编码器 (Digital Video Encoder DENC) SAA7185。它的内部包含了一个同步时钟发生电路和 D/A 转换器。它接受数字化的亮度和色差信号, 编码形成 NTSC/PAL 制的 CVBS 或 S-VIDEO 信号。由于它输入 CCIR601 兼容的视频数据流 (720 像素/每行, 4:2:2 复合格式, 例如 MPEG 译码数据), 因此它直接和 Trimedia 处理器的 Video Out 单元相连, 接受处理器输出的 YUV 复合视频数据流, 它还从处理器接受水平同步、垂直同步信号以及行锁定时钟。同 VIP 一样, 它也包含了 I²C 接口, 通过这个接口从处理器的 I²C 总线上接受控制信号, 来配制和调整它的工作状态。

音频信号输入处理的核心是立体声编码译码芯片 (Stereo Codec) AD 1847, 它可以接受 4 路立体声输入信号, 并且可以提供模拟和数字信号的混合功能, 内部可编程的增益和衰减, 采样频率从 5.5kHz 一直到 48kHz, 串行地输出 16 位或 8 位的数字信号。由于它内部还包含有 D/A 转换电路, 因此它也作为音频信号输出的一部分。由于处理器音频接口的高度可编程性, AD1847 可以无缝地与处理器的 Audio In 单元相连。如果 AD1847 作为串行时钟的从设备, 它从处理器接受采样时钟、字选择信号或帧同步信号, 向处理输出串行的位信号; 如果它作为串行时钟的主设备, 它向处理器输出采样时钟、字选择信号或帧同步信号以及串行的位信号。

音频信号输出部分稍微复杂一些。它由处理器 Audio Out 单元、外部的音频输出控制 D/A 转换器组成, 其中 D/A 转换器包括 AD1847 和 CS4333, 它们都是接受串行输入信号的立体声 D/A 转换器。参考板实现了单独的两路立体声输出和 Dolby 公司的音频标准 (AC-3 六路环绕立体声)。外部的音频输出控制器由可编程逻辑器件 (PLD) 实现, 通过一个 I<sup>2</sup>C 接口电路, 从处理器的 I<sup>2</sup>C 总线上接受控制, 确定当前 Audio Out 输出数据的格式, 然后控制 AD1847 或是 3 片 CS4333 接受串行的比特流信号。通过这样的设计, 使得以 Trimedia 处理器为核心的这个系统既可处理通常的两路立体声信号, 也可处理较新的 AC-3 立体声信号。这就是在前面提到的 Trimedia 的 Audio Out 单元配以少量的外围电路就可实现 8 路立体声; 这也说明通过处理器音频输入输出单元的高度可编程性, 可以使 Trimedia 适应未来的新的音频的标准和协议。

通信处理的核心是电信模拟前端设备 ST7545 (Telcom Analog Front End)。ST7545 是高速的调制解调设备, 完全兼容 V.34 标准, 波特率最高支持到 34800bit/s, 和 Trimedia 的 SSI 接口一起提供了良好的数字调制解调器能力。SSI 接口包含了所有必要的缓冲区和逻辑电路, 因此它们之间的连接是无缝的。

时钟发生电路再简单不过, 由于 Trimedia 处理器结构化设计, 外部只需提供一个 40 或 50MHz (决定于处理器的 CPU 时钟频率, 参考板 CPU 的时钟频率是 80MHz) 的晶振, 通过处理器内部的数字锁相环 PLL (Phase Lock Loop), 倍频得到处理器内部 CPU 时钟, 通过同样方法得到存储系统的时钟。

整个系统的核心是 Trimedia 处理器, 它通过各多媒体输入/输出单元和外围设备连接, 通过 PCI 总线接口和主机的 PCI 总线相连。由于处理器 PCI 总线接口完全兼容 PCI 总线规范 2.1, 因此它和主机之间连接只是增加了一片 5V 到 3V 的电平转换器。处理器具有一个 I<sup>2</sup>C 总线接口, 这样处理器作为 I<sup>2</sup>C 总线的主设备, 而各具有 I<sup>2</sup>C 接口的外围芯片作为从设备, 处理器通过 I<sup>2</sup>C 总线向其他设备发送控制, 外围设备都从 I<sup>2</sup>C 上接受控制, 这样处理器通过 I<sup>2</sup>C 的两条控制线 (SCL 串行时钟线和 SDL 串行数据线) 就完成了通常许多控制线才能完成的控制功能。

从上面的分析中我们可以看出, 由于 Trimedia 处理器设计的结构化和模块化, 使得基于它的开发和应用相对简单, 只需配合少量的外围电路, 就可完成以往需要很多大规模集成电路才能完成的多媒体功能。这不仅降低了设计难度, 减少了开发成本, 而且基于它的应用很容易以低价格占领市场, 实际上这也是当初 Philips 公司开发这款新型媒体处理器的初衷。

(4) 典型的应用。Trimedia 媒体处理器的设计, 使它不仅可作为 PC 系统的加速器,



也可单独应用在独立的系统中。一个典型的应用是 Trimedia 系统作为一个视频解压缩的引擎以 PCI 卡的形式插入在 PC 机里。这时,PC 机无需知道 Trimedia 具有一个强大的通用 CPU,只是把它看成 PCI 卡上的“黑盒子”。PC 机操作系统传递给 Trimedia 处理器一个在 PC 机存储器中的指向压缩视频数据的指针。Trimedia 的 CPU 通过 PCI 总线获得压缩的视频数据流。从视频数据流中解出每帧图像,如果有需要,可以使用 VLD 辅助译码,然后把它们存放在局部的 SDRAM 中。当一帧准备好显示,Trimedia 的 CPU 传给 ICP 一个显示的命令。ICP 自动从 SDRAM 中获得解压缩帧,通过 PC 机的 PCI 总线传递到 PC 机的显示卡的帧缓存中。

Trimedia 的另一个典型应用是视频压缩。在这种情况下,没压缩的视频数据被直接传送到 Trimedia 的视频入单元。一个数字摄像机芯片直接连接到视频入单元,提供复合的 8 位 4:2:2 格式的 YUV 数据。视频入单元从数字摄像机芯片采样视频数据信号,把它们解复合,然后把分离的 Y、U、V 分量分别存放在 SDRAM 中。当完整的视频帧被获取以后,Trimedia 的 CPU 被中断,它将视频数据用软件压缩,然后回写到分离的 SDRAM 中。压缩的视频数据现在可以以多种方式进行处理:可以通过 PCI 总线传送到 PC 机,也可以存放在局部的海量存储器,或者 PC 机把压缩的视频数据流通过网络传输,这些数据也可通过集成的 V.34/ISDN 接口传输到远程系统中,例如视频电话和视频会议系统。当 Trimedia 独立使用时最主要的用途就是设计数字机顶盒。

### 5.3.3 Phenix 芯片和 MMX 技术

多媒体和通信功能集成 CPU 芯片内的另一类,是以通用 CPU 计算功能为主,融合多媒体和通信功能。它们的设计目标是与现有计算机系列兼容,融合多媒体和通信功能,主要用在多媒体计算机中。一种方案是采用标量处理器 (Scalar Engine) 和向量处理器 (Vector Engine) 或称阵列处理机 (Array Processor) 或者是 CPU 和 DSP 相结合的方法,设计全新结构的 CPU。它们的典型产品是 Motorola 公司的 Phenix 芯片。它是把 Power PC 的扩充核标量处理器和向量处理器集成在一个芯片中,称为向量通信处理器 (VECOMP701)。另一种方案是在原有 CPU 基础上扩充多媒体和通信的功能,它们的代表产品是 Sun 公司的 Ultra SPARC-I 和 II, Cyrix Multimedia 586、HP 公司的 MAX-2 及 Intel 公司的 MMX 技术。

#### 1. Motorola 公司 Phenix 芯片

Phenix 芯片把可扩展的 Power PC 的核作为标量处理器和阵列处理机 (Array Processor-AP) 融合在一起,第一代的产品称为向量通信处理器 (VEComP701—Vector Communication Processor 701)。它可以用在信号处理、3D 图形、人机接口以及图像处理等领域。VEComP 701 由两种类型的微处理器组成:一类是单片 32 位 RISC 标量处理;另一类是 32 个 16 位单指令多数据流 (SIMD) 并行操作的向量处理器。VEComP 701 的功能和特性如下:

结构可扩充性:用 VEComP 701 芯片可组成流水线结构,支持  $1 \times N$  个数组,  $N$  最大值是 8。

高性能的标量 RISC 引擎支持设备驱动、中断处理及命令处理:Power PC 单流整数核、32 位地址和数据总线,灵活的数据和指令存储管理、4KB 双信道数据 Cache 地址及 4KB 双信道指令 Cache 地址。

八组存储控制器：到 SRAM, EDO DRAM, EPROM, FLASH 和其他设备的接口、具有位屏蔽的 32 位地址译码。

系统集成单元：时钟合成器、电源管理器、复位控制器、实时时钟寄存器、优先中断时基、硬件总线监控器和软件时基及 IEEE1149.1 JTAG 测试存取信道。

高性能 SIMD 引擎包含 16 个 16 位处理组件：每个处理组件是由专用的 ALU 硬件组成能够同时并行计算数据、每个处理组件有 2KB 专用存储器存放本地数据、1KB 直接映像向量指令 Cache、1KB 直接映像向量数据 Cache、一个向量处理器时钟周期可执行多条指令及高性能的向量输入/输出缓存（VIOBs）。

两个内部的总线：T Bus 和 U-BUS。

总线接口支持工业标准 PCI 协议。

芯片中的 DMA 处理器不需要标量引擎调用可以加速数据传送。

具有调试查错模块。

同步串行接口（SSI—Synchronous Serial Interface）提供用于多种串行设备的全双工的串行通信，例如工业标准的数据压缩算法（Codecs）、DSP 及其他设备。

VEComP 701 的基本结构如图 5-43 所示，它由三个系统组成：标量引擎、向量引擎及外设和系统接口。

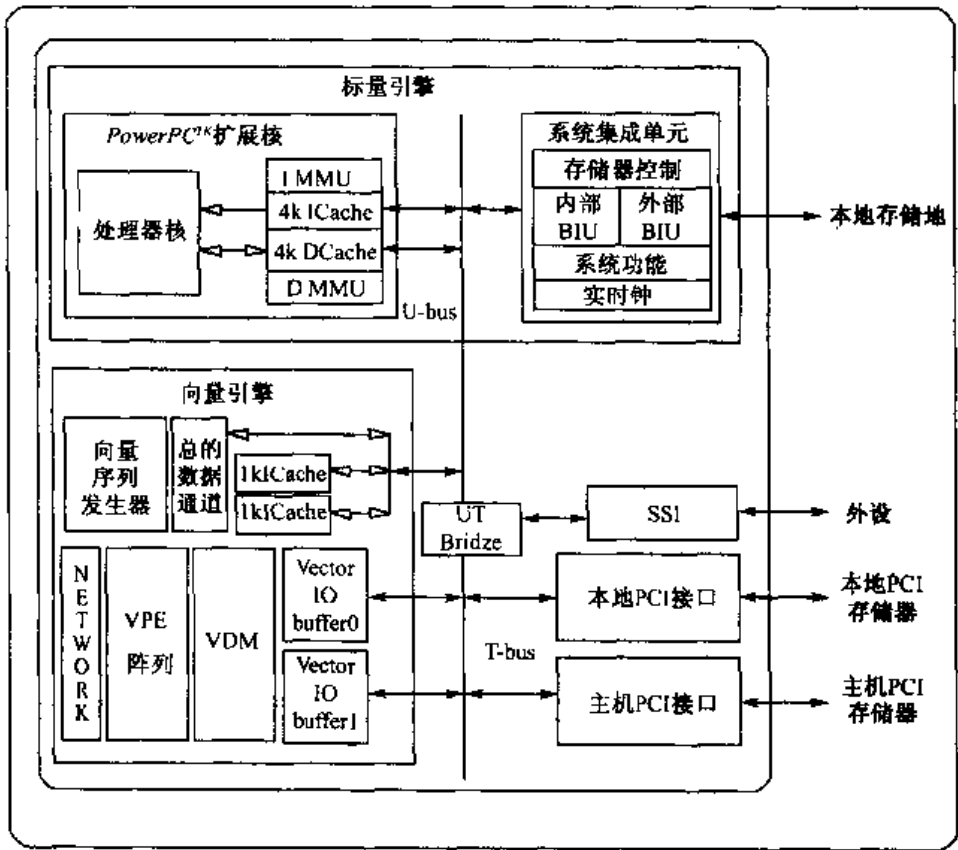


图 5-43 VECOM P 701 结构框图

(1) 标量引擎。标量引擎协调 VECOM P 701 所有的操作，它是由 32 位 Power PC RISC 微处理器组成，它调度下述各种任务：

对于外部数据源的设备驱动器（例如视频帧获取器）。

在板上各种任务的中断服务。

为了被动和监视 SIMD 向量引擎的操作，执行各种通信任务。

为了执行 VECOMP701 各种应用程序，满足数据传送的需要，标量引擎控制多种 DMA 的操作。

1) 标量引擎核心模块在 VECOMP701 内执行 Power PC 的指令系统：

- 执行 Power PC 分支处理器和定点处理的功能。
- 执行 Power PC 用户模式（问题模式）下所有指令，除了浮点指令，相关的特殊指令以及相关的寄存器。
- 支持开发特性、断点和观察点、程序流跟踪数据产生器以及 Debug 模式操作。

2) 标量存储管理。VECOMP 701 标量引擎执行虚拟存储管理方式，它提供 Cache 控制、控制器存取保护以及有效到实际地址的转换。它的操作包括单独的指令和数据存储管理单元（MMU），标量引擎的 MMU 根据 Power PC 操作环境结构（Operating Environment Architecture）进行各种操作。标量引擎的存储管理单元具有：

- 32 项全功能联想数据传送后备缓冲区。
- 32 项全功能联想指令传送后备缓冲区。

3) 标量 Cache。标量数据 Cache 容量为 4KB，两种方式相关物理地址的 Cache。线性容量为 16B，32 位传输信道的存储单元允许每个时钟周期存取 4B 的数据。指令 Cache（IC）容量也是 4KB，两种联想方式的 Cache。Cache 的组织方式共 128 组，每组两行，每行 4 个字。指令 Cache 的行是放在存储器边界的 4 个字。

（2）向量引擎。向量引擎由高性能 SIMD 处理器组成，它由数据驱动，具体地是由向量数据存储单元、向量 VPE（Vector Processing Element）数组、向量序列发生器、向量指令和数据 Cache 以及向量输入/输出缓冲控制器组成的，参考图 5-44 各部分具体情况如下：

1) 向量数据存储单元 VDM（Vector Data Memory）。它为向量引擎提供向量数据存储，它通常存储向量运算的中间结果，或者是将要处理数据的表。它由 16 个存储器块组成，每个块对应 16 个向量处理单元的一个 VPE（Vector Processing Element）。每个存储器的容量是 2048B。由每个 VPE 控制存取这些存储器，同时 VDM 还提供（16 位）半个字及 8 位一个字节的单个存取方式。

2) 向量处理单元数组。向量处理单元数组是由 16 个 16 位处理器单元组成，每个 PE 是由一个专用的算术逻辑部件和用户寄存器功能部件组成。每个 PE 都有一个单周期重加功能的电路，每个 PE 在一条指令中允许 16 个并行操作，同时执行相同的操作。

3) 总的通信信道。为了在向量引擎和标量引擎之间通信，总的通信信道作为公共区域。总的通信信道的源是存放指令操作数和结果的 16 个通用目的寄存器、程序计数器控制向量 PE 的寄存器、屏蔽寄存器、处理重复循环的寄存器以及向量序列发生器的控制状态寄存器等。

4) 向量序列发生器。向量序列发生器相当于通用计算机中的指令寄存器和指令计数器，它完成向量引擎程序线程的取指、译码和执行指令的操作。所有 16 个 PE 都执行相同的操作。

5) 向量指令和数据 Cache。向量 Cache 保存向量引擎所用的数据和指令。这些直接映

像的 Cache 是由 16 个 4 个 16 字节行组成。专用的控制块协调指令未命中和命中 I- Cache 以及数据未命中和命中 D- Cache。

6) 向量引擎输入/输出缓冲控制器。向量引擎输入/输出缓冲区控制器 VIOBC (Vector Engine Input/Output Buffer Controller) 为在 VDM 和外部存储器之间协调 DMA 控制传送时能够提供有效的数据移动。

(3) 外设和系统接口。VEComP 701 用最小的设计复杂性为向量引擎数据传送提供最大的 I/O 带宽, VEComP 701 外设和系统接口包括:

两个内部总线: T 总线和 U 总线。

UT 桥: 它提供 T 总线和 U 总线之间的接口。

测试工具模块: 提供应用程序和芯片功能的测试。

两个向量 I/O 缓存器 (VIOB): 为向量数据存储提供 I/O 控制功能。

直接数据存取 (DMA)。

PCI 接口模块: 为主机和在 PCI 总线上的其他设备提供通信功能。

系统接口单元 (SIU): U 总线和外设之间接口控制。

同步串行接口 (SSI): 为多种串行设备提供全双工串行信道。

1) T 总线。T 总线是 VEComP 701 的内部总线, 为下述 VEComP 701 模块提供通信信道 (参考图 5-43)。

- UT 桥, 到 U 总线的接口。
- 主 PCI 的控制器。
- 本地 PCI 控制器。
- VIOB0 和 VIOB1。

T 总线的特性如下:

- 同序列输出终端同步处理等信号区。
- 采用固定优先级的仲裁方式, 在这种方式中需要一个合理的协议。
- 带有两个先进的地址处理部件的流水线结构总线。
- 具有数据处理功能的终端序列输出。
- 不定长和定长段 (4 个字)。
- 支持一个字节、半个字 (16 位) 及一个字数据宽度。
- 支持数据字节的交换。
- 支持用户定义的地址处理命令 (可任意)。
- 每个时钟分 4 个滴答瞬间。
- 支持每个 T 总线时钟最多传送两个数据字, 在 50MHz 时钟频率时最大总线带宽为 400Mbit/s。

2) U 总线。VEComP 701 的 U 总线是下述几个模块的通信信道 (参考图 5-43)。

- 标量引擎。Power PC 的指令和数据 cache、系统集成单元, 它连接外部总线 (E-BUS) 和本地存储器。
- 向量引擎。指令和数据 cache, 总的通信信道。
- UT 桥。通过它连接到 T 总线。

U 总线的特点如下:

- 具有序列数据终端的同步处理等信号区。
- 具有固定优先级的仲裁方式。
- 具有两个突出地址处理功能的流水线总线结构。
- 具有数据交换功能的不定长和定长段（4 个字）。
- 支持 1 个字节，半个字（16 位）及一个字的数据宽度。
- 50MHz 时钟。

3) UT 桥。UT 桥提供 U 总线和 T 总线的连接，它的功能如下：

- 两个独立的本地存储器 DMA 控制器：LMDMA0 和 LMDMA1，它们可用于调用本地存储器时的所有 DMA 传送，一个可作为 DMA 读，一个作为 DMA 写。
- 用软件寻找 DMA 的源和目的。
- 具有两总线间直接数据传送的队列缓存。
- 具有解决 U 总线和 T 总线内部存取优先级的仲裁逻辑。

4) 测试模块。测试模块用来测试应用软件及芯片中某些特性功能，这个模块是 UT 桥的一部分，它具有 Power PC 扩展核的调试功能，可用于软件开发及选择合适时钟周期的功能。但是它不提供向量引擎和 DMA 的某些功能。它的特性是：

- 窥探缓冲器和跟踪缓冲器能够获取大量信息，它们包括 U 总线周期、T 总线周期、DMA 传送、VE 的指令调度周期以及特殊指令的地址。
- 在规定的 U 总线的地址范围，能够重放窥探缓冲器和跟踪缓冲器获取的信息。外部逻辑分析器能够利用这些信息进行多层次的跟踪分析，这些相同的信息也能够被写到本地存储器中。
- 在每次窥探重放时，时序电路的时间标志加到数据输出的时间戳中，允许测试模块把它用在各种性能分析及软件诊断。
- 具有使用检测各种事件的能力，去控制 VECOMP 701 各个部分，例如停止向量引擎，启动和停止 DMA 部件，中断标量引擎到 Power PC 的诊断模式，可以选择 VE-I cache 的接通和断开并可观看 VE 中 E-BUS 的时钟周期。
- 允许外部逻辑分析器分析输出管脚的信号，并直接控制 VECOMP 701 芯片中的相同部件。在测试模块中所有这些诊断逻辑都连到 UT 桥上。

5) 向量输入/输出缓冲器。VECOMP 701 有两个向量输入/输出缓冲器，它们为向量数据存储器提供控制功能。每个 VIOB 具有独立的操作能力并有专用的 DMA 控制器。向量引擎向量数据存储器通过 256 位数据信道连接到两个向量输入/输出缓冲器（VIOB0 和 VIOB1）。每 VIOB 的 16 行对应一个向量处理单元，VIOB0 和 VIOB1 具有相同的功能。数据输入/输出 VIOB 都要通过 DMA 传送，DMA 传送的数据块是和 VIOB 结构一致，即每次 DMA 写入和读出 VIOB 都要写满和读空。

6) 直接存储器存取。在 VECOMP 701 芯片中直接存储器存取的主要目的是从本地存储器（LMEM1 和 LMEM2）、主机和本地 PCI 总线以及对应 VDM 两个 VIOB 六个 DMA 设备中选择一对作为 DMA 主从设备，进行数据传送。DMA 控制模块为在数据传送期间在两个 DMA 设备之间建立逻辑信道，数据流仅为一个方向。在 DMA 数据信道中的源是主 DMA 设备，目的是从 DMA 设备。VECOMP 701 同时支持 3 个 DMA 信道，这时会有 3 个主设备和 3 个从设备，6 个 DMA 设备在某个瞬间只能是其中之一。

DMA 主从通信可出现在 T 总线, 一个 DMA 的主设备也就是 T 总线的主设备, DMA 的从设备也是 T 总线的主设备, 通信只是单方向的。在 T 总线和 U 总线之间的接口是 UT 桥, 它有两个本地存储器 DMA 信道, 每个都可作为 DMA 的主和从。DMA 数据传送的控制可以通过命令寄存器、状态寄存器及控制块实现, 所有这些寄存器和控制块都在 DMA 主设备和 DMA 从设备中, 在初始化 DMA 传送时, 可以通过软件设置它们。通过不同的 DMA 主从设备, 多个 DMA 传送可以同时发生。

DMA 信道的主设备要设置 4 个控制块, 这些控制块决定数据源的寻址方法, 如起始地址、整个容量、步距及传送字数。从设备也有 4 个控制块, 它决定目的设备的寻址方法。

7) PCI 接口。VEComP 701 有两个 PCI 控制器, 一个主机 PCI 控制器是在主 PCI 总线上, 它是通过主机处理器配置; 另一个是本地 PCI 控制器, 它为外设提供 PCI 总线接口, 它由 VEComP 701 芯片中的标量引擎配置。PCI 接口模块完成在主机同 PCI 总线上的其他设备之间通信所需的所有控制功能, 其特性如下:

- 本地和主机 PCI 接口支持 PCI 总线工业标准协议 2.1 版本 (Version 2.1)。
- 本地 PCI 控制器和主机 PCI 控制器具有相同的功能。

VEComP 701 使用 PCI 总线进行数据传送有下述三种工作方式:

- VEComP 701 作为 PCI 总线的从存储器设备, 这时 VEComP701 要为 PCI 从存储器设置两个独立的存储空间: 从存储窗口和文件区。
- VEComP 701 作为 PCI 总线的主设备。
- VEComP 701 作为 PCI 总线的直接存取设备。

8) 系统接口部件。系统接口部件 (SIU-System Interface Unit) 控制到外部资源的接口以及内部总线的结构。它主要由下述四个功能块组成:

- 系统控制和用户测试。它包括系统时钟发生器、系统复位功能、中断产生器及用户测试功能。
- 内部总线控制。内部总线控制为 U 总线提供多控制管理需要。
- 外部总线控制器。外部总线控制器 E-BUS (External Bus Controller) 提供从 U 总线映像到外部存储器管理控制的各种需要。它的特性是: 32 位地址总线; 32 位数据总线; 为了支持外部和静态存储器设备具有内部产生的片选和等待状态; 不同的存储器类型 (SRAM、EEPROM 及 DRAM); 容易连到从设备接口; 对于总线时钟上升沿的所有信号为同步总线; 支持数据的奇偶性。
- 外部存储器控制器。外部存储器控制器 (EMC, External Memory Controller) 能够使外部总线管理多种外部存储器类型和配置。存储器控制通过通用用户可编程接口达到, 存储器控制特性如下: 连接 SRAM, EDODRAM, DRAM, EPROM, FEPROM, FLASH 及其他外设; 字节写, 可选择的奇偶发生器; 带有屏蔽位的 32 位地址译码。

9) 同步串行接口。同步串行接口 (SSI, Synchronous Serial Interface) 为各种串行设备间提供全双工通信, 包括一个或多个工业标准数据压缩算法 (Codecs)、DSP 及外设。SSI 是由具有 FIFO 缓存及通用 SSI 时钟发生器的独立的接收和发送部份组成。由于所有串行传送对于时钟是同步的, 所以 SSI 接口是同步的。

同步串行接口有三个工作方式: 正常方式、网络方式及按需方式 (On-Demand Mode)。

正常方式是以一定的周期速率传输数据，但是每个周期只传一个字（一个串行数据字可以是 8 位，16 位或 24 位）。像正常方式一样，网络方式也是以固定周期传送数据，但是它支持每个周期传送最多 32 个字，这种方式可以用在网络上的多任务器。按需方式是非周期的传送数据，这种方式可以高速串行传输数据。

SSI 控制器可以作为 UT 桥模块的一个部分运行于 U 总线的主设备或从设备。对于正常和按需方式，标量引擎软件能够直接存取发送和接收的 FIFO。对于网络方式，SSI 可以作为总线的主设备，为了提供线路发送和接收缓冲区可以使用本地存储器。SSI 的特性如下：

- 时钟频率最高为 16.6MHz（对应 50MHz 通用系统时钟）。
- 可以使用外部串行时钟源或提供时钟源。
- 字长可为 8 位、16 位或 24 位。
- 具有单独的接收和发送 FIFO，每个容量为 5 个 32 位字。
- 具有发送和接收功能的独立帧。

总结 VECComp701 的工作原理如图 5-44 所示。

## 2. Intel 公司的 MMX 技术

1996 年 3 月 5 日 Intel 公司首先对外公布了 MMX 技术，它是由设在以色列海法的 Intel 实验室完成的，1997 年 1 月 9 日 Intel 公司对外正式推出含有 MMX 技术，具有多媒体扩充指令集的多能奔腾处理器 P55C，1997 年 5 月 Intel 又进一步推出了具有 MMX 技术的 P6 奔腾芯片，主频可达 300MHz 的 Pentium II 300，进一步提高了性能。对计算机市场产生了较大的影响，尤其是对多媒体计算机市场，一些 PC 机厂家认为 MMX 技术会对多媒体硬件支撑平台带来根本的变化，纷纷推出基于 P55C 和 P6 的多媒体计算机。有些厂家认为 MMX 技术是解决计算机、电信和家电三电一体化的较好的方案，因此积极投入人力和物力研制开发三电一体化的新产品；而另一些多媒体板级产品的厂家也在利用 MMX 技术设计制造新的多媒体板级产品以适应市场的需要。总之，MMX 技术的出现，已对多媒体计算机市场产生了深远的影响。

(1) MMX 技术的设计思想。为了改善 Intel 体系结构 IA (Intel Architecture) 的多媒体和通信性能以及适应其数字信号处理的应用，Intel 公司采用 MMX 技术扩充 IA。为了保持和扩大 Intel 公司已有 CPU 芯片的国际市场，MMX 技术保证向下兼容性。MMX 技术利用 SIMD（单指令多数据流）技术开发了很多算法内部蕴藏的并行机制，所以它能比没有 MMX 技术的 CPU 芯片运行速度快。

MMX 技术的开发者分析了大量多媒体和通信技术的应用软件，包括图形、MPEG 视频压缩、音乐合成、音频压缩编码和译码、图像处理、游戏、语音识别以及视频会议，发现虽然它们是不同的应用领域，但是在数据类型和计算方法方面有共性，它们只有简单的数据类型（8 位的像素 RGB 或 YUV 及 2, 16 位的声音采样数据），定点的矩阵向量运算、局部的循环以及高度的并行性。采用 SIMD 结构正好能够在一条指令中并行执行多数据流相同的操作，这就是 MMX 技术能够加速的最根本原因。

另一个设计思想是使 MMX 技术与现有 Intel PC 机的操作系统和软件全兼容，因此对 MMX 的设计不得不上许多限制，如不能引进新的状态寄存器、控制寄存器及新的条件码等，所以设计者使用浮点寄存器作为 MMX 的寄存器组。这样设计的原因是：同为浮点寄存

器能移提供 64 位字长，可作定点 8 位、16 位或 32 位并行运算；另一个原因是保证了与现有应用软件与操作系统的全兼容性。

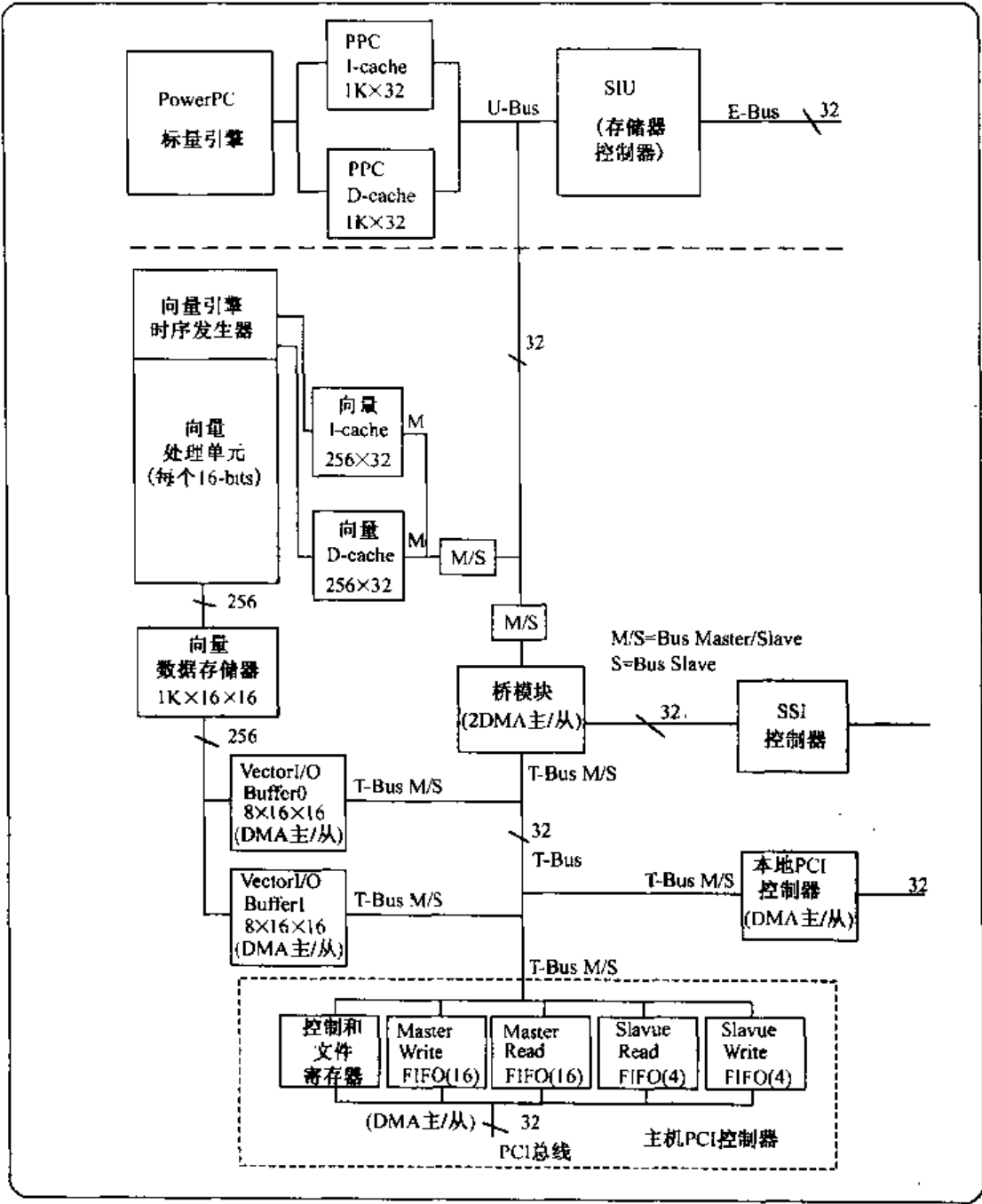


图 5-44 VECComp701 详细的原理方框图

(2) Intel MMX 的核心技术。MMX 技术提供了面向多媒体和通信应用的新特性，同时保持了对全部现有 Intel 体系结构微处理器、IA 应用程序和操作系统向下全部兼容，新特性如下：

- 增加了新的数据类型。把较小的数据元素的数据类型合并到一个寄存器中。
- 扩充了饱和型运算方式。定点运算时上溢下溢不中断，保持最大最小值。
- 扩充的 57 条新指令。扩充的 MMX 指令系统采用 SIMD 形式完成寄存器中所有数据元



素的并行操作。

与 IA 结构的全兼容性。8 个 64 位 MMX 寄存器组可映像到 IA 结构的浮点寄存器中。

1) 新的数据类型。音频采样数据是 16 位字长，灰度图像的像素数据是 8 位字长，彩色图像或图形的 RGB 各分量也是 8 位字长。如果在一个 64 位的数据信道中同时处理 4 个 16 位数据或 8 个 8 位数据，运算速度可以提高到原来的 4 倍或 8 倍，同此 MMX 技术定义了三种打包的（或称紧缩的）数据类型及一个 64 位字长的类型。在打包的数据类型中每个元素都是定点整数。四种数据类型定义如下（参考图 5-45）：

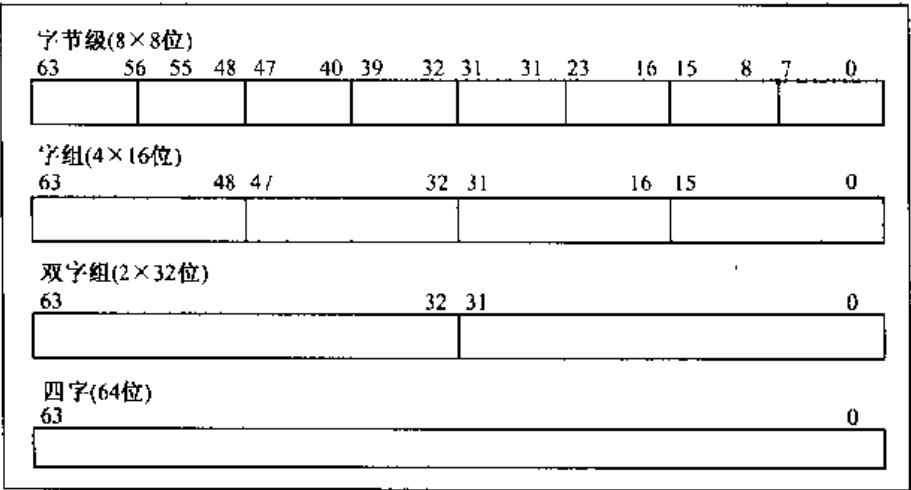


图 5-45 成组数据类型

- 字节组类型：8 个字节组组成的一个 64 位数据。
- 字组类型：4 个字组成的一个 64 位数据。
- 双字组类型：2 个双字组成的一个 64 位数据。
- 四字类型：1 个 64 位数据。

MMX 技术可以在一条指令中同时处理 8 个、4 个或 2 个数据，所以称它为单指令多数据流（SIMD）并行处理结构

2) 扩充的饱和型运算方式。定点运算经常会遇到运算结果上溢或下溢。对有符号数运算，上溢指的是结果超过正的最大数，下溢指的是结果比负的最小数还要小。当浮点处理器遇到这种情况时，通常是作为异常情况处理，设置溢出标志并产生中断，由系统软件处理，或者停机交程序员处理，因为这时产生的结果是完全错误的。在图形和图像处理运算中，经常用 8 位无符号数表示一个像素值或彩色分量值，其最大值 255 表示亮度最大值（最亮），如果运算结果超过这个值仍然表示为最亮，则是符合实际情况的。但实际上由于溢出进位使其数值变小，图像变黑或变灰，导致与实际情况不符。MMX 技术采用饱和式运算方式（Saturation Mode），当运算结果达到最大值时便不再增加，而是保持在这个值。这样就减少了溢出判断处理所需的内部操作而加快了运算速度。另一方面，饱和运算不是一种特殊的操作模式，也不用设置寄存器，它是某些指令操作码的一部分，只在加、减指令中才有饱和方式。例如，对于 8 位有符号数，上溢和下溢分别置成 7FH 和 80H；8 位无符号数置成 FFH 和 00H。

3) 扩充的 57 条新指令。扩充的丰富的 MMX 指令系统可以将多种数据元素（8×8，4

×16 或 2×32 位定点) 编组成 64 位进行并行操作。共有 57 条 MMX 指令增加到原有的 IA 中, 表 5-4 给出了这些指令的类型及对它们的简单描述。

表 5-4 MMX 指令系统简表

类 型	助 记 符	不同操作码个数	描 述
算术	PADD[B, W, D]	3	带绕转加
	PADD[S][B, W]	2	带饱和的符号数加
	PADDUS[B, W]	2	带饱和的无符号数加
	PSUB[B, W, D]	3	带绕转减
	PSUB[S][B, W]	2	带饱和的符号数减
	PSUBUS[B, W]	2	带饱和的无符号数减
	PMULHW	1	压缩型字高位乘
	PMULLH	1	压缩型字低位乘
	PMADDWD	1	压缩型字相乘并相加结果对
比较	PCMPEQ[B, W, D]	3	压缩型数相等比较
	PCMPGT[B, W, D]	3	压缩型数大于比较
转换	PACKUSWR	1	字压成字节 (无符号数带饱和)
	PACKSS[WB, DW]	2	字压成字节, 双字压成字 (有符号数带饱和)
	PUNPCKH[BW, WD, DQ]	3	把 MMX 寄存器高阶展开
	PUNPCKL[BW, WD, DQ]	3	把 MMX 寄存器低阶展开
逻辑	PAND	1	按位与
	PANDN	1	按位与非
	POR	1	按位或
	PXOR	1	按位异或
移位	PSLL[W, D, Q]	6	压缩型逻辑左移 (位数由立即数或 MMX 寄存器指定)
	PSRL[W, D, Q]	6	压缩型逻辑右移 (位数由立即数或 MMX 寄存器指定)
	PSRA[W, D]	6	压缩型算术右移 (……)
转移	MOV[D, Q]	4	数据转码
状态	EMMS	1	清 MMX 状态

从表 5-4 中可见, 可以把新增加的 57 条 MMX 指令分成算术运算指令、比较运算指令、转换运算指令、逻辑运算指令、移位元运算指令、数据转移指令、MMX™ 状态置空 (EMMS) 指令。

(a) 算术运算指令:

数据组的加减:

PADD (数据组加) 和 PSUB (数据组减) 指令对有符号或无符号数据元素进行加减, 按环绕式处理方式将源操作数加到目标操作数上或从目标操作数中减掉源操作数。这些指令支持字节组、字组和双字组。

PADD[S] (数据组饱和加) 和 PSUB[S] (数据组饱和减) 指令对有符号或无符号数据元素进行加减, 按饱和处理方式将源操作数加到目标操作数上或从目标操作数中减掉源操作数, 并按数据类型的取值范围对运算结果进行饱和处理。这些指令支持字节组和字组。

PADDUS (无符号数据组饱和加) 和 PSUBUS (无符号数据组饱和减) 指令对无符号数

据元素进行加减，按饱和处理方式将源操作数加到目标操作数上或从目标操作数中减掉源操作数，并按数据类型的取值范围对运算结果进行饱和处理。这些指令支持字节组和字组。

数据组的乘法：

数据组的乘法指令对 16 位无符号操作数按对应关系进行了四次乘法，产生了 32 位的中间结果。用户可以选择任何一个 32 位结果的高位或低位部分。

PMULHW（数据组高位字乘法）和 PMULLW（数据组低位字乘法）指令将有符号的源操作数和目的操作数的字（Word）相乘，并将每个结果的高 16 位或低 16 位写入目的操作数。

数据组的乘加：

PMADDWD（数据组相乘并相加）指令将源操作数和目的操作数的有符号字部分相乘，并将所产生的四个 32 位双字的中间结果成对相加，产生两个 32 位双字。

（b）比较指令：

PCMPEQ（数据组相等比较）和 PCMPGT（数据组大于比较）指令将源操作数和目的操作数对应的数据元素分别进行相等比较或大于比较。这些指令在目的操作数产生或 0 或 1 的屏蔽。逻辑操作可以根据这些屏蔽选择数据元素。它可以无需单个分支指令或一组分支指令就能实现有条件的赋值操作。指令不设置标志位。

这些指令支持字节组、字组和双字组的数据类型。

（c）转换指令：

成组和分组指令在成组数据类型中执行类型转换的功能。

PACKSS（有符号饱和处理模式的类型转换）指令按饱和处理模式，将有符号字组转换成有符号的字节组或将有符号的双字组转换成有符号的字组。

PACKUS（无符号饱和处理模式的类型转换）指令按无符号饱和处理模式，将无符号字组转换成无符号的字节组。

PUNPCKH（成组数据高位分组）和 PUNPCKL（成组数据低位分组）指令可将字节组转换成字组、字组转换成双字组或将双字组转换成四字组。

（d）逻辑指令：

PAND（逐位逻辑与），PANDN（逐位逻辑与非），POR（逐位逻辑或）和 PXOR（逐位逻辑异或）指令在 64 位运算中逐位执行逻辑运算操作。

（e）移位元指令：

逻辑左移、逻辑右移和算术右移指令按指定的位数移动每个数据元素。逻辑左移和右移也能够使 64 位（四字）数据作为一个数据块进行移动，辅助数据类型的转换和对齐操作。

PSLL（逻辑成组左移）和 PSRL（逻辑成组右移）指令执行逻辑左移逻辑右移，并用 0 填充空出来的高位或低位部分，这些指令支持字组、双字组和四字组数据类型。

PSRA（数据组算术右移）指令执行一个算术右移，并将符号位复制到操作数的高端。该指令支持字组和双字组数据类型。

（f）转移指令：

MOVD（转移 32 位）指令将成组的 32 位从存储器移到 MMX™ 寄存器或逆向移动数据，或从整型寄存器中移到 MMX™ 寄存器或逆向移动数据中。

MOVQ（转移 64 位）指令将成组的 64 位从存储器移到 MMX™ 寄存器或逆向移动数据，

或在 MMX™ 寄存器中进行数据移动。

(g) EMMS (清 EMMS 状态) 指令:

EMMS 指令将 MMX™ 状态清空。该指令必须在一个 MMX 例程结束时执行。用于清除 IA MMX™ (清空浮点标志字) 使下一个例程能够执行浮点运算指令。

除 EMMS 指令外, 所有的 MMX™ 的指令操作均涉及到两个操作数, 即源操作数和目的操作数。右边的操作数为源操作数, 左边的操作数为目的操作数。目的操作数也可作为第二个源操作数来使用。指令用结果来覆盖目的操作数。

例如, 一个双操作数的指令可以按如下方式译码:

DEST (左边操作数) ← DEST (左边操作数) OP SRC (右边操作数)

对所有的 MMX™ 指令来说 (除数据转移指令), 源操作数可以位于存储器和 MMX™ 寄存器。目的操作数寄存于 MMX™ 寄存器中。

对于数据转移指令, 源操作数也可以是一个整型寄存器 (对 MOVD 指令) 或是存储器单元 (对 MOVD 和 MOVQ 指令)。

4) 与 IA 结构的全兼容性。MMX 技术提供了 8 个 64 位通用寄存器 (如图 5-46 所示), 实际上它们是 IA 中的浮点寄存器, 在 MMX 技术指令中可以通过 MM<sub>0</sub>-MM<sub>7</sub> 直接对它们进行寻址操作。IA MMX™ 寄存器状态是建立在 IA 浮点寄存器状态上的别名, 在 MMX™ 技术中没有增加新的状态和模式。对浮点状态进行存取的浮点指令同时也处理了 IA MMX™ 状态 (例如, 在进行上下文切换时)。MMX 技术在浮点体系结构和操作系统之间使用了相同的接口技术 (主要用于任务切换的功能)。

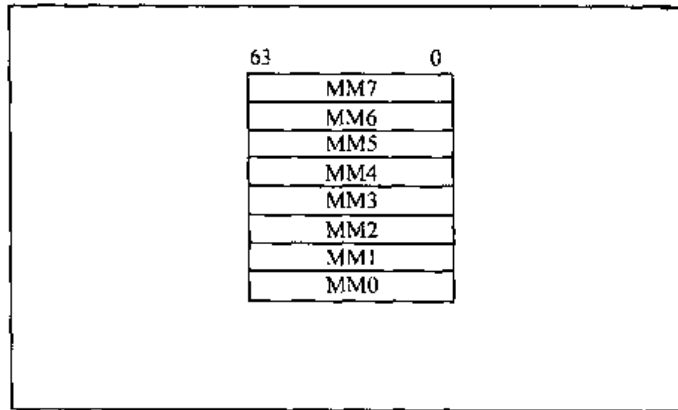


图 5-46 MMX™ 寄存器集合

(3) MMX 技术与奔腾处理器体系结构。在这一节我们将介绍具有 MMX 技术的奔腾处理器与 P6 系列处理器如何执行 MMX 指令流, 从中可以了解到为什么这种优化能够提高代码的运行速度, 这会帮助我们调度和优化应用程序, 以求得更高的运行速度。

1) MMX 奔腾处理器。奔腾处理器是一个超标量的处理器, 它具有两个通用的流水线和一个可流水作业的浮点单元。奔腾处理器在一个时钟内可完成两条指令, 一个流水线完成一条指令。一个流水线称为 U 流水线, 一个称为 V 流水线。在任何一条给定的指令译码期间, 将检查后面的两条指令, 如果有可能第一条指令被安排到 U 流水线, 而第二条安排到 V 流水线。具有 MMX 技术的奔腾处理器为通用流水线增加了额外的处理阶段, 具有 MMX 技术奔腾处理器中 MMX 指令流如图 5-47 所示。

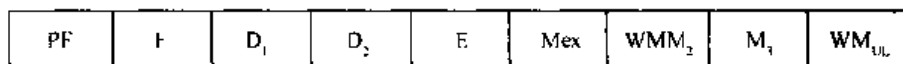


图 5-47 MMX 指令流

奔腾处理器的流水线只有 E，它可以在两条流水线中放有 E<sub>1</sub>、E<sub>2</sub>……，而具有 MMX 技术的奔腾处理器指令从代码的高速缓冲区取出后，放到 PF 预取阶段，并在 F 提取阶段对预取的指令字节进行语法分析，对有前缀的指令译码成多至两条指令，并压入 FIFO。如果每条 MMX 指令不超过 7 个字节则可压入两条指令，指令在先进先出缓冲区中将语法分析与指令译码分开，这个缓冲区位于 F 阶段与译码 1 (D<sub>1</sub>) 阶段之间。整数、浮点和 MMX 指令在 D<sub>1</sub> 流水线阶段译码。D<sub>2</sub> 阶段为读取源操作数。E 阶段是执行指令阶段。Mex 阶段为 MMX 指令的执行阶段，ALU 成组移位元和分组在读时钟内完成。乘法指令所需的第一时钟阶段，无阻塞条件。WMM<sub>2</sub> 阶段为单时钟指令写入阶段，乘法器的第二个时钟阶段，无阻塞条件。M<sub>3</sub> 为乘法器的第三个时钟阶段。WM<sub>UL</sub> 是写乘法器结果的操作阶段，到此一段 MMX 指令全部完成。

2) P6 系列处理器。P6 系列处理器使用动态执行结构。该结构通过硬件寄存器重命名和分支预测的方法，将乱序执行和推测执行合成一起。这些处理器有一个有序进入的流水线，它将宏指令分解成简单的微操作和一个可以处理这些微操作的乱序的超标量处理器内核。这个乱序的处理器内核包含了几条流水线，连接了整数、跳转、浮点和存储器执行单元。几种不同的执行单元可以集成在同一流水线上。大多数简单操作（整数 ALU，浮点加法，甚至浮点乘法）可以按每时钟周期完成一至两个操作的吞吐量进行流水作业。

P6 系列的流水线由三部分构成：有序组织的前端 (In-Order Issue Front-end) 单元，乱序内核 (Out-of-Order Core) 单元和有序的退出 (In-Order Retirement) 单元。

由于动态执行处理器按乱序的方式执行指令，所以产生了数量充足的、可供执行的微操作，并使大多数有关性能调节方面的考虑得以实现。正确的分支预测和快速的译码是有序前端单元提高性能的核心。

在每个时钟周期内，指令译码阶段可以对多达 3 条的 Intel 体系结构宏指令进行译码。但是，如果指令复杂或指令长度超过 7 个字节，译码器的译码指令数将有所下降。

译码器可以译码：

- 每个时钟周期多达三条宏指令。
- 每个时钟周期多达六条微操作。
- 指令长度大于 7 的宏指令。

P6 系列的处理器在 D<sub>1</sub> 阶段上有 3 个译码器。第一个译码器可在每个时钟周期完成一个由四个以下微操作构成的宏指令，其他两个译码器在每个时钟周期内完成一个由一个微操作构成的宏指令。由多于 4 个微操作构成的指令将耗费多个时钟周期来完成译码，在使用汇编语言编程时，微操作序列来安排指令将增加每个时钟周期内的译码指令数。通常：

- 简单的寄存器。寄存器格式的指令仅用一个微操作。
- 读取指令仅为一个微操作。
- 存储指令仅为一个微操作。
- 简单的读—修改指令为两个微操作。

- 简单的寄存器。记忆体格式的指令由 2~3 个微操作构成。
- 简单的读—修改—写指令由 4 个微操作构成。
- 复杂的指令通常超过 4 个微操作，故需耗费多个时钟周期译码。

一旦微操作被译码，它们从有序前端单元发送到保留站中，保留站是乱序核心管道段的开始部分。在保留站中，微操作等待它们的操作数变为有效。一旦一条微操作的所有操作数有效，它将从保留站中被送到一个执行单元。如果一个微操作在数据准备好的状态下进入保留站（即为全部数据有效），那么该微操作将立即被送到一个合适的且有效的执行单元中。在这种情况下，在保留站中的微操作只耗费了很少的时钟周期，全部的执行单元聚集在保留站的输出埠上，一旦微操作执行完将返回到全排序缓冲区 ROB，并等待退出。在这个管道阶段中，全部数据值被回写到存储器并且全部的微操作按序退出，一次三条。

3) 高速缓存 (Cache)。具有 MMX 技术的处理器的在片高速缓存子系统，由两个 16K 的 4 路线长为 32 字节的关联高速缓存体构成。高速缓存具有一个回写机制和一个伪 LRU 的置换算法。数据的高速缓存由 8 个按四字节边界交错的存储体构成。

在具有 MMX 技术的奔腾处理器上，只要引用的数据不在同一个高速缓存体上，就可以被两个管道同时访问。在动态执行的 (P6 系列) 处理器上，只要引用的数据不在同一个高速缓存体上，就可以被一条读取指令和一条存储指令同时访问。在具有 MMX 技术的奔腾处理器上，高速缓存访问失败的延迟为 8 个内部时钟周期。在具有 MMX 技术的动态执行处理器中，最小延迟是 10 个内部时钟周期。

4) 分支目标缓存。具有 MMX 技术的奔腾处理器和动态执行处理器在分支预测方面，除一个较小的异常处理外，在功能上完全一样。

分支目标缓冲区 (BTB) 存储了预先所见的分支和它们的目标。当一个分支被预取后，BTB 将目标地址直接填入到指令读取单元 (IFU)。一旦分支被执行，BTB 将随着目标地址而改变。使用分支目标缓存时，预先所见的分支被动态预告。分支目标缓存的预测算法包括了模式匹配和每目标多达 4 位的预测历史位。例如，一个具有 4 个迭代长度的循环将百分之百地被正确预测到。

编写条件分支 (除循环外) 可将最常执行的分支紧接在分支指令后 (即失败)。

另外，具有 MMX 技术的处理器有一个堆栈返回缓存 (RSB, Return Stack Buffer)，可以连续地为不同地址上调用的过程，正确地预测其返回地址，进一步为展开具有函数调用的循环带来了益处，并删除了某些需要 in-line 的过程。

在具有 MMX 技术的奔腾处理器上，如果两个分支指令的最后一个字节在同一个按四字节对齐的存储器段内，则分支不可预测。

这种情况，发生在两个相连分支间没有间隔指令且第二个指令只有两个字节长的情况下 (如  $\pm 128$  字节的相对跳转指令)。

为避免这种无法预测的情况，应使第二分支加长，在分支指令中用 16 位的相对位移代替 8 位的相对位移。

5) 写缓存。具有 MMX 技术的处理器具有 4 个写缓存 (相对无 MMX 技术的奔腾处理器的两个写缓存)。另外，写缓存可以被 U 管道使用，也可以被 V 管道使用 (相对无 MMX 的奔腾处理器的一个写缓存对应一个管道的情况)。通过对存储器写操作进行安排调度，可以提高关键循环的性能。如果你不想看到写未命中，每组指令不能安排多于 4 条写指令。并

在安排另外的写指令前调度其他指令。

(4) MMX 开发工具和编程技巧。由于 MMX 技术在汇编语言级增加了新的指令和数据操作类型, 因此要想使用 MMX 指令必须熟悉这些指令和它们的格式。另外, MMX 新的指令是可以同时处理和存储多个数据的 SIMD 并行处理指令。必须严格按指令的要求去组织数据流, 才有可能充分发挥 MMX 的潜力。我们所熟悉的并行语言(如并行 C), 一般是针对多 CPU 环境下的多指令多数据流(MIMD)并行处理, 而 MMX 技术是指令级并行 RISC 超标量结构, 不能用并行 C 实现并行任务分配与处理。在现有情况下, 建议采用下述工具和方法。

1) 采用在高级语言中嵌入 MMX 指令的方法。微软的 VC++4.2 和 Powersoft/Watcom C 是最早以嵌入式汇编支持 MMX 扩展的 C 语言编译器。Intel 公司自己开发了几几乎与 MMX 指令相对应的 C 指令库, 可以避免记忆繁琐的汇编格式, 减少寄存器分配等操作, 但是数据流的组织仍然要由程序员完成。

2) 采用 Intel 公司提供的 MMX 标准函数库。Intel 公司现在已经提供了用 MMX 指令系统实现的、面向多媒体应用的标准函数库: 数字信号处理库(DSP)和模式识别库(PRL)。这些库函数包含了多媒体应用中一些常用的操作, 如快速傅立叶变换(FFT)、离散余弦变换(DCT)、特征抽取等。这两个库函数较新的 V3.0 版本是用 MMX 指令实现的, 只要利用 6.11D 版本的宏汇编语言连接这两个库, 就可以利用 MMX 指令。值得注意的是 DSP 和 PRL 这两个库函数, 无论是其标准函数的数量还是执行速度都在随着 MMX 技术的发展及用户的增加而不断改善。

3) 采用数据流描述方法。传统的算法语言是建立在计算机结构控制流模型上, 然而在计算机算法描述中, 控制流的描述并不是本质的, 例如两个向量的加法既可用循环结构写成, 也可以用顺序结构写成, 还可以写成混合的折衷形式, 而所有这些程序的执行结果都是等效的。保证这些程序功能相同的关键是数据流, 也就是说算法的功能被其数据流唯一地确定。基于这种思想, 设计了一种数据流描述方法, 把 MMX 的并行性能描述成可以让平行数据流并行通过的数据信道。用这种方法可以很方便地将传统程序转换为 MMX 程序, 并且便于调试。

MMX 技术是为了扩充 CPU 的多媒体和通信功能而设计的, 所以它为实现多媒体和通信应用程序带来了方便。最近 Intel 公司给出了一些应用 MMX 指令集编程的示例, 如彩色键连、向量点积、矩阵乘法、淡入淡出等。我们也用 MMX 指令集在图形图像领域开发了一些应用程序, 例如, RGB 到 YUV 彩色空间的转换、中值滤波及图像边缘提取等, 运算速度可以提高 3.5~5 倍。

MMX 主要的编程技巧分述如下。

1) 寻址方式的选择。在奔腾处理器上, 当一个寄存器被用作基地址元素时, 如果该寄存器是前一个指令的目的寄存器(假设所有的指令都已在预取队列中), 将耗费一个附加的时钟周期。例如:

```
add esi, eax      ; esi 是目的寄存器
mov eax, [esi]    ; esi 是基地址, 增耗费 1 个时钟
```

因为奔腾处理器有两条整数流水线, 如果一个寄存器是前一时钟内任意指令的目的寄存

器，这个用于计算有效地址（在任一管道）的基地址或索引元素的寄存器，将耗费一个额外的时钟周期。这种效应称为地址生成互锁（AGI）。为了避免 AGI，指令间应安排其他指令，并产生至少一个时钟周期的间隔来分隔这些指令。

新增的 MMX™ 寄存器不能当作索引寄存器或基地址使用，所以 AGI 不适用于 MMX™ 寄存器为目的寄存器的情况。

在 AGI 情况下，动态执行（P6-系列）处理器不产生额外延迟。

2) 数据和代码的对准。奔腾处理器和动态执行的 P6 处理器都有一个 32 字节行长的 cache。由于预取缓冲区在 16 字节边界进行操作，代码对准对于预取缓冲区的效率有直接的影响。

对于在 Intel 体系结构下的优化，推荐以下几点：

- 如果循环入口标记距离下一个 O MOD 16 边界小于 8 字节，则应将这一入口标记置于该边界处。
- 条件转移后面的标记不必对准。
- 非条件转移或函数调用后面的标记应该进行与第一种情况相同的对准。

在 Pentium 中，数据 cache 或总线中的非对准访问至少要多消耗 3 个时钟周期。在动态执行处理器（Pentium Pro 系列）中，数据 cache 的非对准访问要额外消耗 9~12 个时钟周期。为了在所有的处理器中都得到最好的执行效率，Intel 建议数据要进行如下的对准：

- 2-byte 数据要全包含在对准的 4-byte 字中（即其二进制地址应为 xxxx00, xxxx01, xxxx10, 但不可以是 xxxx11）。
- 4-byte 数据要对准于 4-byte 边界。
- 8-byte 数据（即 64-bit，例如双精度实数据类型，所有的 MMX 紧缩寄存器数据等）必须对准于 8-byte 边界。

3) 有前缀的操作码。在 Pentium 处理器中，指令的前缀会延迟指令分析（parsing）过程和禁止指令的配对执行。下面给出了指令前缀在 FIFO 中影响：

- 以 0Fh 为前缀的指令没有以上不良影响。
- 以 66h 或 67h 为前缀的指令要消耗 1 个时钟周期检测前缀，1 个周期做长度计算，1 个周期进入 FIFO（总共 3 个时钟周期）。它必须是第一个进入 FIFO 的指令，后面的指令可以和它一起推入。
- 其他前缀（除了 0Fh, 66h, 67h 外）的指令要消耗 1 个多余的时钟周期来检测每个前缀。这些指令仅作为第一个指令时被推入 FIFO 中。一个有两个前缀指令要消耗 3 个周期推入 FIFO 中（前缀的处理需要 2 个周期，指令要 1 个周期）。在同一时钟周期中，下一个指令可以和第一个指令一起推入 FIFO 中。

仅当 FIFO 中只有一项时，才会产生性能上的降低。只要译码器（D1 状态）得到两个指令进行译码，就不会有性能的降低。如果从 FIFO 中以每个时钟周期两个指令的速度弹出指令，FIFO 会迅速变空。这样，如果带前缀指令之前的指令遇到性能缺失（例如，没有指令配对，由于 cache 不命中引起的阻塞，存储没有对准等），则带前缀指令的额外开销就会被屏蔽掉。

在动态执行（P6）处理器中，长于 7 字节的指令会限制每个周期译码的指令数。前缀会增加 1~2 字节的指令长度，从而限制了译码器的性能。



建议尽量不要使用带前缀的指令，或者将它们安排在会引起流水线阻塞的指令后面。

4) Pentium Pro 系列处理器中的寄存器部分阻塞。在动态执行处理器 (P6) 中，如果在写 1 个 16 位或 18 位寄存器 (例如, AL、AH、AX) 之后紧接着读 1 个 32 位寄存器，读操作将阻塞，直到写操作完成 (至少需要 7 个周期)。考虑如下的例子，第一条指令将立即数 8 写入 AX 寄存器，第二条指令访问了 EAX 寄存器。这段代码就会导致寄存器部分阻塞：

```
MOV AX, 8
ADD ECX, EAX
```

由于 Pentium Pro 系列 CPU 可不按严格的顺序执行代码，所以不是紧密相连的指令也会引起阻塞。下面的例子就包含了一个寄存器部分阻塞：

```
MOV AL, 8
MOV EDX, 0X40
MOV EDI, new-value
ADD EDX, EAX
```

在第四条指令访问 EAX 时会发生寄存器部分阻塞。

注意，Pentium 处理器没有以上的阻塞情况。

5) 配对。在所有的 Intel 处理器中，“配对”都是提高执行效率的有效方法。下面给出在 Pentium 和 P6 上进行配对的一些原则。在一些情况下，对于特定处理器的性能优化有一些折衷，这些折衷基于应用的特定性质而有所不同。在超标量的 Pentium 处理器中，指令的顺序对于最大限度地利用性能是十分重要的。

对指令顺序的重排，产生适当的配对，可以增加并发执行 (利用 U-流水线和 V-流水线) 两个指令的可能性。有数据相关的指令之间至少要相隔 1 个其他指令。

下面将讨论 MMX 指令与整数指令的配对问题，浮点指令是不能和 MMX 指令配对的。

- 如果两个指令中有 1 个长于 7 字节，Pentium 处理器就不能给它们配对。但是，MMX 技术的 Pentium 处理器仅当第一个指令长于 11 字节或第二个指令长于 7 字节时无法配对。
- 带前缀的指令在 U-流水线中可以配对。带 0Fh, 66h 或 67h 前缀的指令也可以在 V-流水线中配对。
- 都要使用 MMX 移位元单元 (pack, unpack 和 shift 指令) 的指令不能配对，因为 MMX 只有一个移位元单元。移位操作在 U-流水线或 V-流水线都可以进行，但是不能同时在 1 个时钟周期中进行。
- 都要使用 MMX 乘法单元 (pmull, pmulh 和 pmadd 等指令) 的指令不能配对，因为 MMX 只有一个乘法单元。情形同前。
- 访问存储器或整数寄存器堆的 MMX 指令仅可以在 U-流水线中使用。所以，这样的指令不能被配对。
- U-流水线指令的 MMX 目标寄存器和 V-流水线指令的源寄存器或目标寄存器不相同，才可以配对。不然，这种相关阻碍并发执行。
- EMMS 指令不能配对。

此外还有动态分支预测，静态预测，高速缓存优化及存储器优化等编程技巧。

总之，MMX 程序的开发是一项技巧性很强的工作。MMX 技术的利用需要程序员具有良好的汇编语言编程能力，对 MMX 技术要有深入的理解，同时，还需要在实践中不断总结经验，才能写出高效的代码来。

## 5.4 小结

多媒体计算机是综合处理声、文、图信息。为了更好发挥多媒体计算机的功能，必须有高性能的硬件支持。视频和音频的专用处理芯片和新型的体系结构是多媒体计算机硬件支持的关键。

本章详细介绍了具有代表性的 Intel/IBM 公司研制的数字视频交互式多媒体计算机系统 DVI 系统的组成和工作原理、Intel/IBM 公司设计的 AVSS 和 AVK 软件，以及 DVI 系统的成功和失败的经验教训。

为了更好在解决多媒体计算机综合处理声、文、图信息。提出了三种处理方案：

- 选用专用芯片设计专用接口卡单独解决。
- 设计专用芯片和软件组成多媒体计算机系统。
- 把多媒体和通信功能集成到 CPU 芯片中。

本章对 Mpct 媒体处理器和 Trimedia 媒体处理器的工作原理以及开发应用也作了介绍和讨论。

## 5.5 习题

1. 组成多媒体系统的途径有哪些？

- (1) 直接设计和实现。
- (2) 增加多媒体升级套件进行扩展。
- (3) CPU 升级。
- (4) 增加 CD-DA。

- (A) 仅 (1) (B) (1) (2)
- (C) (1) (2) (3) (D) 全部

2. 下面硬件设备中哪些是多媒体硬件系统应包括的？

- (1) 计算机最基本的硬件设备。
- (2) CD-ROM。
- (3) 音频输入、输出和处理设备。
- (4) 多媒体通信传输设备。

- (A) 仅 (1) (B) (1) (2)
- (C) (1) (2) (3) (D) 全部

3. MPC-2、MPC-3 标准制定的时间分别是\_\_\_\_\_。

- (1) 1992 (2) 1993
- (3) 1994 (4) 1995

- (A) (1) (3) (B) (2) (4)  
(C) (1) (4) (D) 都不是

4. 下面哪些是 MPC 对音频处理能力的基本要求?

- (1) 录入声波信号。  
(2) 处理声波信号。  
(3) 重放声波信号。  
(4) 用 MIDI 技术合成音乐。

- (A) (1) (3) (4) (B) (2) (3) (4)  
(C) (1) (2) (3) (D) 全部

5. 下面哪些是 MPC 对视频处理能力的基本要求?

- (1) 播放已压缩好的较低质量的视频图像。  
(2) 实时采集视频图像。  
(3) 实时压缩视频图像。  
(4) 播放已压缩好的高质量高分辨率的视频图像。

- (A) 仅 (1) (B) (1) (2)  
(C) (1) (2) (3) (D) 全部

6. 下面哪些是 MMX 技术的特点?

- (1) 打包的数据类型。  
(2) 与 IA 结构完全兼容。  
(3) 64 位的 MMX 寄存储器组。  
(4) 增强的指令系统。

- (A) (1) (3) (4) (B) (2) (3) (4)  
(C) (1) (2) (3) (D) 全部

7. 下面哪些称得上是多媒体操作系统?

- (1) Windows98 (2) Quick Time  
(3) Avss (4) Authorware

- (A) (1) (3) (B) (2) (4)  
(C) (1) (2) (3) (D) 全部

8. 下面哪些是 MPC 对图形、图像处理能力的基本要求?

- (1) 可产生丰富形象逼真的图形。  
(2) 实现三维动画。  
(3) 可以逼真、生动地显示彩色静止图像。  
(4) 实现一定程度的二维动画。

- (A) (1) (3) (4) (B) (2) (3) (4)  
(C) (1) (2) (3) (D) 全部

9. 详述 Inte/IBM 公司研制的 DVI 多媒体计算机系统成功和失败的经验教训? 理想的系统如何设计和实现?

## 5.6 参考答案

1. (B)
2. (C)
3. (B)
4. (D)
5. (A)
6. (D)
7. (C)
8. (A)
9. (略)

## 第6章 超文本和超媒体

### 本章要点

- 超文本和超媒体的基本概念及其主要特点和基本特性。
- 超文本和超媒体的体系结构，分为三个层次：表现层—用户接口层；超文本抽象机层—节点和链；数据库层—存储、共享数据和网络访问。
- 超文本和超媒体的文献模型。
- 超文本和超媒体的应用以及超文本和超媒体存在的问题和发展前景。

在当今的信息社会里，信息以爆炸形式增长，影响着人类的工作、学习、生活各个方面。面对浩如烟海且相互关系复杂的信息，使得人们感到现有的信息存储与检索机制越来越不足以使信息得到全面而有效的利用，尤其不能像人类思维那样以通过“联想”来明确信息内部的关联性，而这种关联却可以使人们了解分散存储在不同地点的信息块之间的联系关系及相似性。因此，迫切需要一种技术或工具，它可以建立起存储于计算机网络中信息之间的链接结构，形成可供访问的信息空间，使得各种信息能够得到更广泛的应用。

随着多媒体计算机技术的发展，超文本和超媒体也蓬勃地发展起来。这种现象的出现不是偶然的。除了技术上的进步、市场的需要以外，另一个重要因素就是超文本和超媒体较适合于表达多媒体信息。多媒体技术使得计算机表达信息的方式更丰富、更直观，同时，提供了灵活的人机交互手段。多媒体表达信息的方式主要有两类：第一类在时间轴上对多媒体信息进行编辑和剪裁；第二类在空间上安排多媒体信息，共同表达事物，做到“声、文、图”一体化。超文本和超媒体提供了将“声、文、图”结合在一起综合表达信息的强有力手段。同时，作为一种接口技术，超文本和超媒体提供了非常直观、灵活的人机交互方法，开拓了许多应用领域，是多媒体应用的有效工具。

随着多媒体技术的发展，超文本和超媒体技术以其能为人们提供简单、直观、快捷、灵活的数据表示、组织和管理手段而展现出美好的前景。它所提供的思想方法可建立各种媒体信息之间的网状链接结构，与传统的线性文本结构有着很大的不同，符合人们的“联想”式思维习惯。

### 6.1 概述

#### 6.1.1 基本概念

文本是我们最熟悉的信息表示方式。文章、程序、书、文件等都以文本出现，通常以字、句子、段落、节、章作为文本内容的逻辑单位，而以字节、行、页、册、卷为物理单位。文本的最显著特点是它在组织上是线性的和顺序的。这种线性结构体现在用户读文本

时只能按固定的线性顺序一字一字、一行一行、一页一页地读下去。文本的这种线性结构如图 6-1 所示。

科学研究表明，人类的记忆是一种联想式的记忆，它构成了人类记忆的网络结构，人类记忆的这种联想结构不同于文本的结构。人类记忆的互连网状结构可能有多种途径，不同的联想检索必然导致不同的访问路径。例如，某人对“通信工具”一词可能会联想到“手机”，对自己童年的照片可能联想到小时候的快乐生活，尽管我们对某一对象具有相同的概念，但由于文化基础和受教育的背景，由于不同时间或不同的地点，产生联想的结果就可能是千差万别的。

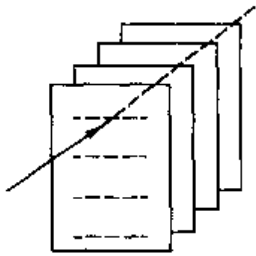


图 6-1 文本的线性结构

这种联想方式实际上表明了信息的结构及其动态性。显然，这种互连的网状信息结构用普通的文本是无法管理的，必须采用一种比文本更高一层次的信息管理技术，即超文本（hyper-text）。

超文本结构就类似于人类的这种联想记忆结构，它采用一种非线性的网状结构组织块状信息，没有固定的顺序，也不要求读者必须按某个顺序来阅读。采用这种网状结构，各信息块很容易按照信息的原始结构或人们的“联想”关系加以组织。例如，一部百科全书有许许多多“条目”，它可以按照字母次序进行排列，也可以按照各专业分类用链加以连接，以便于人们“联想”查找。图 6-2 是一个完整的小型超文本结构。从图中可以看到超文本是由若干内部互连的文本块（或其他信息）组成，这些信息块可以是计算机的若干屏，也可以是若干窗口、文件或更小信息块。这样一个信息单元就称为一个节点（node）。不管节点有多大，每个节点都有若干指向其他节点或从其他节点指向该节点的指针，这些指针被称为链（link）。链有多种，它连接着两个节点，通常是有向的，从一个节点（称之为源节点）指向另一个节点（称之为目的节点）。链的数量通常不是事先固定的，它依赖于每个节点的内容和信息的原始结构。有些节点与其他节点有许多关联，因此它就有许多链；有些节点没有启程链，它就只能作为目的节点。超文本的链通常连接的是节点中有关联的词或词组而不是整个节点。当用户主动点触该词时将激活这条链，从而迁移到目的节点。

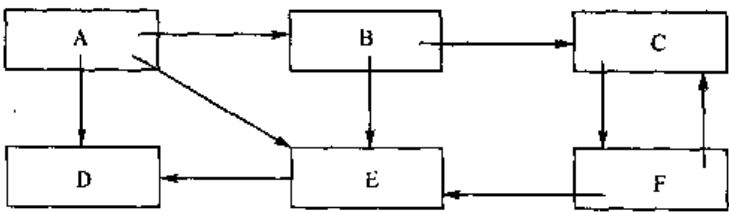


图 6-2 具有 6 个节点和 9 条链的超文本结构示意图

图 6-2 所表示的超文本结构实际上就是由节点和链组成的一个信息网络，称为 Web。读者可以在这个信息网络中任意“航行”浏览。这里要强调的不仅仅是“阅读”，而更重要的是用户可以主动地决定阅读节点的顺序。假如读者是从标记为 A 的文本块开始阅读，与单一路径的文本不同，该超文本结构有三条阅读路径摆在读者面前，即可到 B、D、或 E。如果读者选择 B，则可以继续选择到 C 或 E，从 E 又可以到 D。当然读者也可以从 A 选择直

接到 D。这个例子表明, 在超文本结构中任意两节点之间可以有若干条不同的路径, 读者可以自由地选择最终沿哪条路径阅读文本。这同时要求超文本结构的制作者事先必须为读者建立一系列可供选择的路径, 或者由超文本系统动态地产生出相应的路径, 而不是单一的线性路径。

超文本的主要特点可以概括为超文本不是顺序的, 而是一个非线性的网状结构, 它把文本按其内部固有的独立性和相关性划分成不同的基本信息块, 称为节点, 以节点作为信息单位。一个节点就可以是一个信息块, 也可以是若干节点组成一个信息块。超文本具有以下特点。

### 1. 多媒体信息

超文本的基本信息单元是节点, 它可以包含文本、图形、图像、动画、音频和视频等多种媒体信息, 而且它的信息表现方式和大小等都可根据所要表述的主题自由选择、组合, 不需要严格的定义。

### 2. 网络结构形式

超文本从整体来讲是一种网络的信息结构形式, 按照信息在现实世界中的自然联系以及人们的逻辑思维方式有机地组织信息, 使其表达的信息更接近现实生活。

### 3. 交互特性

信息的多媒体化和网络化是超文本静态组织信息的特点, 而交互性是人们在浏览超文本时最重要的动态特征。

能对超文本进行管理和使用的系统叫超文本系统。超文本与超文本系统的关系和数据库与数据库管理系统类似。一个超文本系统一般具有以下特点:

(1) 在用户界面中包括对超文本的网络结构的一个显示表示, 即向用户展示节点和链的形式。

(2) 给用户一个网络结构的动态总貌图, 使用户在每一时刻都可以得到当前节点的邻接环境。

(3) 超文本系统一般使用双向链, 这种链应支持跨跃各种计算机网络, 如局域网和因特网。

(4) 用户可以通过自己思维的联想及感知, 根据自己的需要动态地改变网络中节点和链, 以便对网络中的信息进行快速、直观、灵活的访问, 如浏览、查询、标注等。这种联想和感知被准确地定义, 并要求有良好的性能/价格比。

(5) 尽可能不依赖它的具体特性、命令或信息结构, 而更多地强调它的用户界面的“视觉和感觉”。

多媒体信息与传统的单媒体信息有很大的不同, 多媒体的表现是具有特定含义的, 它是一组与时间、形式和媒体有关的动作定义。先进的多媒体表现的交互特性可提供用户控制表现过程和存取所需信息的能力。多媒体信息的组织将有助于信息的表达和交互。多媒体和超文本的结合大大改善了信息的交互程度和表达思想的准确性。多媒体的表现又可使超文本的交互式界面更为丰富。

早期超文本系统的表现形式仅是文字的, 这就是它被称为“TEXT”的原因。随着多媒体技术的发展, 各种各样的多媒体接口的引入, 表达信息的形式扩展到用视觉和听觉甚至触觉来表现。近年来, 随着图形、硬件、大容量存储等技术的发展产生了第二代超文本系统,

它的节点信息可将正文、图形、声音、动画、静止图像、活动图像结合在一起,因而更具魅力。第二代超文本系统与多媒体技术结合起来,为强调处理多媒体信息的能力而称为超媒体(Hypermedia)系统,即多种媒体的超文本系统。

### 6.1.2 典型的超文本系统

## 1. HyperCard

1987年8月,美国苹果公司在 Macintosh 微机上推出了 HyperCard 软件,这是一个十分形象的集图文声为一体的超文本系统,它的出现对超文本技术的推广起着重大作用。

HyperCard 的基本信息单元为卡片, 相当于节点, 第一个版本上一张卡片占有一个屏幕, 它可由用户通过工具来制作。这类工具有编辑器、图符库、剪辑板、绘图工具等。一张卡片有其固定背景部分, 包括一些按钮、可填充的域及图形边框; 一张卡片还有其本身的卡片名、索引号、域内容、正文及图形内容等。一组卡片由用户组成堆 (Stack), 如“产品”堆卡片可包括公司的许多不同产品。卡片堆以索引号排序, 堆的主人可对卡片堆进行各种操作: 加一张卡片, 修改某卡片内容, 删一张卡片, 按一域的内容重新排序卡片, 按某字符串查找卡片, 往前一张卡片, 向后一张卡片, 返回第一张卡片, 显示最近看过的卡片清单; 等等。

各堆卡片可组成堆的目录，目录也是一张卡片，可用浏览工具漫游各个卡片堆。在 HyperCard 2.0 版本中已构造了许多应用卡片堆，如电话本、日历、备忘录、产品目录、价格表、图片集、相片集、看图识字本等。在 HyperCard 软件中用户可交互地制作软件，也可通过该软件提供的一个面向对象语言 Hypertalk 来编写脚本，脚本附于按钮，按钮出现在 HyperCard 堆的卡片里。图 6-3 展示了 HyperCard 环境的概念结构。由于 Macintosh 具有许多方面的工具，因而 HyperCard 制作、浏览均十分方便灵活。HyperCard 的链只有顺序、重新排序等，比较简单，因而连接复杂关系时尚支持不够。但是，1987 年首先推出的这一商品软件以它丰富多彩的界面、灵活方便的构造工具，向广大计算机用户展示了超文本的生命力。

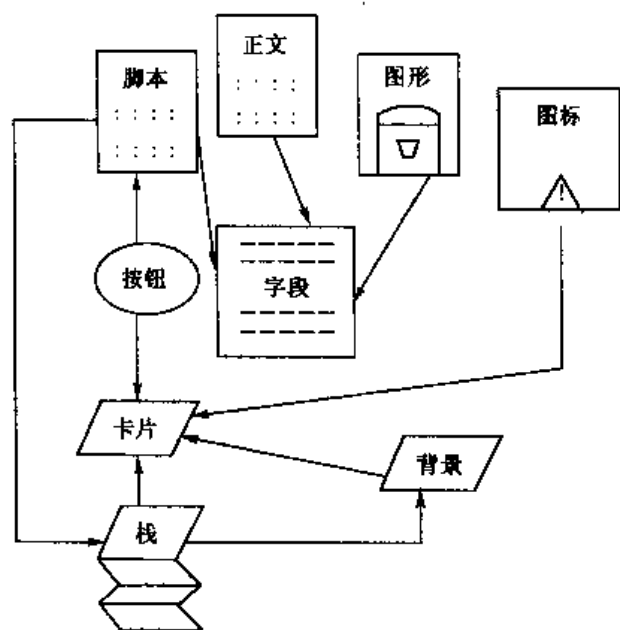


图 6-3 HyperCard 环境的概念结构



## 2. KMS

KMS 是卡内基—梅隆大学在工作站上开发的“Knowledge Management System”的简称，它是该校于 1972 年开始研究的 ZOG 项目在工作站上的新版本。它作为一个通用工具，已用于联机手册、录像带盘的管理、飞机专家系统的界面及交互任务的管理等。KMS 的基本信息单元为帧（frame），最大占用一个屏幕。一帧有一个名字、一个标题、一个正文或图形描述以及多指向其他帧的标志。例如，一篇文献的第一页构成一帧，文章的标题及摘要作为该帧的标题及内容，文章的各节点名、关键字、交叉引用、注释等均可作为指向相应帧的标志（链源，也称 hotword）。在 KMS 屏幕上有一组通用菜单项，如 Save, Reset, Prev, Next, Home, Goto, Info, Disp, Exit, 用户可对它进行所需要的操作。

## 3. Note Card

NoteCard 是美国 Xerox 公司的 PARC 研究中心为研究设计人员或作家开发的一个基于工作站的超文本系统。NoteCard 有一个良好的浏览工具，包括一个层次系统和组织复杂的 NoteCard 网络。

NoteCard 与 Augment (Engelbert, 1984) 的区别在于它基于网络而不是基于联机构成。NoteCard 的基础是语义网络，这是按链归类互连的 NoteCard 集合。NoteCard 提供了一组工具：包括对于网络的组织、显示和管理，还有为创建管理此网络的程序而提供的方法和协议。

NoteCard 包含以下几个基础实体：

(1) 记事卡 (NoteCard)。一张电子模拟的索引卡片。每一张记事卡都有一个标题和可以被编辑的内容资料，例如正文和图形。

(2) 链 (Link)。把单个的记事卡连成相关卡片的网络。每一链是一个在源卡片和目的卡片之间的有向连接。

(3) 浏览器 (Browser)。它含有 NoteCard 网络的结构图。在 NoteCard 系统中，浏览器一般被视作支持编辑的工具。用户可以通过对在 Browser 表中的节点和边的操作来编辑网络结构。

(4) 文件盒 (Filebox)。它帮助用户管理 NoteCard 的巨大网络。文件盒可以用于建立层次目录结构，这种结构对独立于网络互连的卡片提供有效的存储和检索。

NoteCard 的主要特点可归纳如下：

在 NoteCard 系统中核心结构是一个语义网络，它由电子记事卡组成，这些电子记事卡通过分类链将节点连接起来。每一记事卡含有一任意数量的信息，具体表现为正文、图形、映像或某些其他可编辑的信息；每一个链设计为在两个记事卡之间的一种专门的关系，这种关系可以由用户或系统来定义。系统提供浏览器和文件盒这两种特殊类型的卡片，以帮助管理卡片和链的网络。一个浏览器含有卡片网络的结构图，它可以被系统生成或被用户生成，浏览器是一描述工具和编辑机制。文件盒可以用于聚集和组织记事卡，产生由系统管理的分层文件结构。NoteCard 还包括一组协议和功能，以创建新的卡片类型和操纵在网络中的信息。这种面向应用的扩展在公式化表示中起着重要作用。

## 4. Guide

Guide 系统是 1986 年由办公工作站有限公司 (OWL) 推出的，它推出时即为商品化软件，所以是第一个广泛应用的超文本，可以在 Macintosh, IBM PC 等多种个人机上运行。

Guide 系统比 HyperCard 和 NoteCard 更面向正文，它有许多不同的链类型：

(1) 替换链 (Replacement Links) 使当前窗口的文本用该链的目的节点中的正文所代替。在文本中, 一个链的实体通过光标位置的改变而被指明。用户也可以改变显示的文本 (使用黑体、斜体), 以增进对不同类型链的识别。

(2) 记事链 (Note Links) 弹出显示资料正文的窗口。

(3) 引用链 (Reference Links) 引用带有目的正文的新窗口。这些链允许用户在不同资料或同一资料的不同部分中的相关题目之间移动。

(4) 查询链 (Inquiry Links) 类同于表示链, 但这种链是互斥的。在某一时刻, 对某一查询只有一种表示可以被显示在屏幕上。

## 6.2 超文本和超媒体系统的组成和结构

### 6.2.1 超文本和超媒体的组成要素

#### 1. 节点

超媒体是由节点和链构成的信息网络, 节点是围绕一个特殊主题组织起来的数据集合, 这个集合可以是有形的, 例如是一个数据块; 它可以是无形的, 是信息空间中的一个部分。我们可以把一篇文章分割成若干块, 这些块就是有形的节点。若对文章不进行分解, 而只是根据需要对相应内容进行定位, 则这个定位周围的信息就是一个无形的节点。节点中可以嵌入链, 使它能与其他节点相连接。

节点有许多种, 而且分类方法也不尽相同。在早期超文本中节点的内容一般是有形的节点, 内容主要是文本、符号或数字。现在根据媒体的种类、媒体的内容和功能的不同, 节点可以是媒体节点, 其中可以包含各种媒体, 也可以包含数据库、文献等; 也可以是动作类节点、组织类节点、推理类节点等。

(1) 媒体类节点。媒体类节点中存放各种媒体信息, 包括文本、图像、图形、视频、动画等各种媒体, 也包括数据库、文献, 存放这些媒体信息的来源、属性和表现方法等。在一些情况下, 每一个节点中确实包含媒体数据本身, 但也有一些情况特别是在网络环境下, 许多媒体数据需要临时从网络或机器中得到, 所以节点中只有路径、属性等信息, 而没有数据本身。

节点中对媒体数据的描述直接关系到多媒体数据的表现, 不同的媒体会有不同的属性和表现方法。例如, 对文本要能够表现出文本的字体、排版和大小; 对图像来说要能够指明位置和大小; 对视频要能够定义诸如快进、暂停之类的操作; 对数据库这种结构化的数据要能具有符合数据库操作的手段。对混合媒体来说, 媒体之间的同步、配合和效果就要有更复杂的描述形式。

(2) 动作与节点。动作与操作也是一种媒体, 因此可以当作一种动态节点, 它通过超媒体的按钮来访问, 所以有人也称之为按钮节点。在这种节点中常常定义了一些操作, 例如电话通信、传真等, 通过这种节点为用户提供动作和操作的可能。例如有些超媒体系统就专门提供了电话通信功能, 只要用户选择一个“自动拨号盘”的按钮, 就可以开始一次通话。还有一些超媒体系统将传真服务引入, 并与电视通信相结合, 用户按下“传真”按钮, 系统就在当前节点上发送所需传送的内容。实际上这类节点是通过按钮做一些超媒体表现以外的

工作，赋予人的操作或动作。但要注意，动作和操作并不是非要专门的节点，可以嵌入到任何节点中，按钮也一般都与链相连接，只不过动作和操作的按钮连接的是执行链。

(3) 组织型节点。组织型节点是组织节点的节点。加索引是描述节点的一种方法，同时也是数据库管理的需要。组织型节点可以实现数据库的部分查询工作，如结构查询。组织型节点包括各种媒体节点的目录节点和索引节点。目录节点包含各个媒体节点的索引指针，指向索引节点。索引节点由索引项组成，索引项用指针指向目的节点，或指向相关的索引项，或指向相关表中相对应的一行，或指向原媒体的目录节点。

(4) 推理型节点。推理型节点用于辅助链的推理与计算，包括对象节点和规则节点。推理型节点的产生是超媒体智能化发展的产物。

需要指出的是，现代的许多超媒体系统中有的已经没有了节点的概念，或者说节点已经无形了。也有的系统将节点分为原子节点和组合节点，原子节点是不能再进一步分割的对象，如标志、图元、背景、表格的字段等；组合节点由原子节点构成，例如文本的词或段落可以是原子节点，文本就是组合节点。但我们了解了节点的有形概念，还是有利于对超媒体的理解。

## 2. 链 (link)

链又称超链 (hyperlink)，是节点间的信息联系，它以某种形式将一个节点与其他节点连接起来。由于超媒体没有规定链的规范与形式，所以分类方法也不尽相同。但最终达到的效果却是一致的。即建立起节点之间的联系。链是有向的，一般结构可分为三个部分：链源、链宿及链的属性。链源是导致浏览过程中节点迁移的原因，可以是热标、媒体对象或节点等。链宿是链的目的所在，可以是节点，也可以是其他任何媒体内容。链的属性决定链的类型。

由于各个超媒体系统的链型不完全一样，这里仅介绍一些比较典型的链型，供读者参考。

### (1) 基本结构链

基本结构链是构成超媒体的主要链形式，它具有固定明确的特点，必须在建立一个超媒体文献时事先由作者指明，是一种实链。基本结构链又包括基本链、交叉索引链和节点内注释链。

1) 基本链。它是建立节点之间基本顺序的链，这有些类似于一本书中具有 的章、节、小节、段落等结构。它使信息在总体上呈现出层次结构。如图 6-4 中的实线所示。基本链的链源和链宿都是节点。在表现时常 用“上一节点”、“下一节点”等来表现节点的先后顺序，也即链的方向。

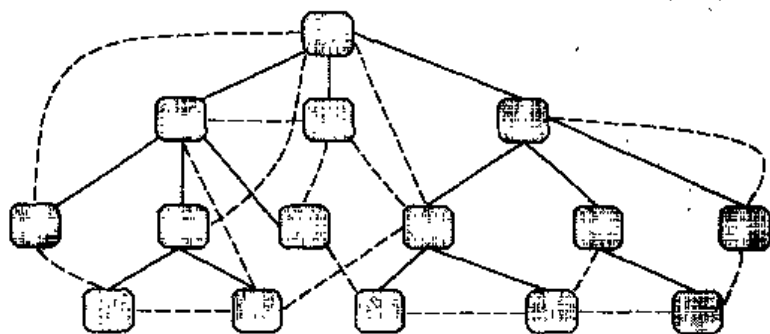


图 6-4 节点中的基本链和交叉索引链

2) 交叉索引链。它将节点连接成交叉的网状结构,如图 6-4 中的虚线所示。交叉索引链的链源可以是各种热标、单媒体对象及按钮,链宿为节点或任何内容。在表现时常常用热标激活转移、“回退”、“返回”等表示先后顺序,要注意的是,这些操作基本链与交叉索引链是不同的,基本链的动作决定节点间的固定顺序,而交叉索引链的动作决定的是访问顺序。

3) 节点内注释链。它是一种指向节点内部附加注释信息的链,注释源主要通过热标确定,注释体则为一单媒体对象。之所以称其为节点注释链,是因为链源和链宿均在同一节点内,一般这种节点都是混合媒体节点。采用节点内注释链的好处是不用另设节点,在需要时注释才出现。在表现形式上,注释需要对热标进行激活才能动作。

### (2) 组织链和推理链

组织链用于节点的组织,推理链则在链的迁移过程中通过推理来决定目标。

1) 索引链。它将用户从一个索引节点到该节点相应的索引入口。索引用于文献与数据库的接口及查找共享同一索引项的文献,按钮表现常是“总目录”、“影片索引”等。

2) 执行链。执行链将一种执行活动与按钮节点相连。执行链使应用程序不再是孤立的,可以激发一个动作或操作。一般的操作系统无法记录程序的功能、目的等,但超媒体的按钮节点与执行链可以通过建立节点方便地解释应用程序的功能和目的,使超媒体成为高层程序的界面。

推理链将在后面专门介绍。

### (3) 其他链型

1) 自动链接。自动链接是超媒体中一个非常重要的概念,它允许系统自动把当前节点与相似主题或满足某些条件的所有其他节点链接在一起。例如可以在文本文件中搜索关键词并报告关键词所在的行数,或是通过基于内容检索确定某些特征,或是通过特殊的通信协议与外部的服务器中的内容建立联系等。但目前能够实现的主要是文本节点间的自动链接,实现其他类型媒体节点间自动链接的超媒体系统还未见到。自动链接的另一个意义在于对超媒体进行基于内容的检索,检索时输入某些主题、特征或条件,系统将能满足该主题、特征或条件的节点自动链接起来,提交用户浏览,这样就大大减少了用户在信息海洋中无用的操作。

2) 类型链。有些超媒体系统允许用户描述在于两个节点之间的关系,也即用户可以定义链的类型。例如:节点 B 给出一个例子,来说明节点 A 中的规则或原理,连结 A、B 的链就称为“注释”链。但假设存在另一个节点 C,它对节点 B 有一个更常用的解释,可以称“B、C”之间的链为“扩展”链。对于这种类型的链必须要用一个独立的数据实体来描述。链实体独立于节点,因为任何给定的节点都可以用不同的方式与其他节点链接。采用类型链的好处在于,首先,在一条链迁移之前,用户可以预先知道目标节点的自然属性。其次,从理论上讲,它允许用户对节点进行预查询。另外有了类型链,开发智能超媒体就大大方便了,这是因为关于知识最重要的方面是事实、假设或规则之间的关系。单个的事实、假设或规则都不能说明问题,除非知道它们之间是怎样相互关系的。如果超媒体网络中存储了关于各种节点之间关系的知识,则可以查询这些知识,例如可以询问:“显示所有讨论某问题或与之类似问题的节点”,“显示包含与某问题相反论据的节点”等。

### 3. 热标 (hotspot)

热标是确定信息关联的链源, 由它将引起向相关内容的转移。很显然, 不同的媒体应有不同的形式的热标。根据媒体种类的不同, 热标的形式一般有以下几种:

(1) 热字 (hot-word)。热字是文本中被指定具有特殊含义或需进一步解释的字、词或词组。图 6-5 便是一例。

在图中, 斜体加底线的词都是热字, 触发这些词将会按照设计者的安排出现相应的进一步的解释, 或出现更形象的演示, 或转移到另外相关内容显示。例如, 触发“商品价格”一词, 可以转移到另一段关于“商品价格”的更详细的说明, 触发“图片”将会出现一幅图片等。

对于热字的处理关键是热字的识别和按要求进行转移。一个字或词究竟是不是热字, 热字如何转移, 都要由设计者进行定义。有的系统用特殊的符号标识热字, 凡是热字一律用保留字括起, 并指明转移的方向或处理的方式。下面是一些不同类型的例子:

本商城的商品价格是按照国家有关规定制定的, 所售出的商品在其规定的质量保证期内可以到本商城退换或者进行维修, 查阅商品信息可以获得该商品的更多介绍。选择图片可以看到商品的式样。

商城在发展电子商务方面主要是面向客户 (B-C), 目前我们也正在开展 (B-B) 业务。

图 6-5 带热字的文本

.....@图片|picture.bmp@所示...  
.....@商品价格|^ (节点名称)^@...  
.....@商品信息|^ (文献名)^节点名@...  
.....@ (B-C) |^9@... (转到第 9 段)

其中, ‘@’ 之后 ‘|’ 之前的词为热字, 在 ‘|’ 之后则是相应的转移目的地, 后一个 @ 为结束符。在实际显示时, 各个保留字及转移目的地等均不显示, 热字被赋予特别的颜色, 所以仍然可以保持原有媒体的显示风格, 并且很容易地与一般的文本编辑器相兼容。转移的目的地与转移的处理方法与超媒体系统本身的设计有关。

(2) 热区 (hot-area)。热区是在所显示的图像或类似于图像的显示区上指明的一个敏感区域作为触发转移的源点。在一幅图像上的不同区域可以有不同的信息表现。例如一幅人体图像中的不同区域可以设置成不同的热区, 当触发这些热区时, 系统就会按设定好的方法进行表现, 介绍该人体部位的详细情况和细节。热区的设定不同于热字, 由于图像十分直观但不便于用语言或文字描述, 所以一般都采用所见即所得的方式在图中直接指定热区。早期的热区一般使用矩形, 但当敏感区域为复杂边缘时, 矩形会造成较大的误差, 所以近来一般都改用多边形。事实上, 当图像都是文字时, 也可以用热区的方法模拟热字, 但此时系统并不知道热区中的字或词是什么, 文本本身也无法滚动; 对文本的修改会引起图像位置的改变, 从而使热区也不准确; 同时也不利于对文本的查询。热区在触发后所引起的转移与文本中的

热字相同，所不同的是文本热字必须在文中描述转移的目的地，而热区则需要在生成时指明并存储在节点的链中。

(3) 热元 (hot-element)。在图形媒体中，图元是其最基本的单位，例如一个图、一条线、一串文字等。为了使这些相对独立的图形单位能够作为信息转移的链源，就引入了热元的概念。这种方式非常适合于在不影响图形本身的变换（例如移位、放大或缩小）的同时，又可以由该图元引发相应的进一步关联信息的表现。例如，在军事态势图上，某图标代表某个部队，该部队的进一步情报都由该图标触发。当图标移位后（如部队调动），仍能引发相应的信息。这用热区、热字都是无法做到的。同样，热元也可以用在 CAD 工程设计中作建筑图注释、机器设备联机维护手册等方面。由热元而导致的转移与热区相似。

(4) 热点 (hot-point)。热点是另外一种热标的概念，主要用于时基类媒体如动态视频、声音等在时间轴上的触发转移。在应用中常常出现这种情况，例如当用一般视频在介绍某个重大历史事件过程中，往往突然会对其中某个片段更感兴趣，从而希望了解更多的内容。这就要求能从这段视频的相应时间轴处转移到另外有关解释的其他内容处，这个起点处就称为热点。在这一点上它与文本媒体十分相似，帧序列可以像文本段一样在序列内、文献内或文献间进行转移。视频对象可以采用长序列，要由起始帧和结尾帧确定所选定的视频段，从而可以从一个视频段直接跳往另外一个视频段，也就可以实现自我解释。其他时基类媒体也基本相同。

由于时基类媒体是动态的，在使用时不能仅将热点定为时间上的某一时刻，因为用户很难准确的确定这一时间点。热点应是一个由用户设定的时间区间。热点如果定于  $a$ ，则在识别时应给出一个  $[b, a, c]$  的敏感区间，在此区间内的触发都应算作有效。由于时基类媒体有“表现—理解”的滞后效应，往往在理解了某一段内容后才可能有了解它信息的愿望，而此时该时刻已过。为了正确地对应，热点区间亦往后对应，一般至少区间  $[b, a]$  要远小于区间  $[a, c]$ ，以适应该滞后效应。

(5) 热属性 (hot-attribution)。这是把关系数据库中的属性作为热源来使用。由于关系框架下的各元组可以根据操作产生许多不同的结果，比如不同的排序顺序、选择不同元组子集等，但总的来说，数据媒体是一种特定的格式化符号数据，所以大多数情况下可以采用类似于热字的热标方法。热标源单位一般为一个属性，用特定的保留属性字的方法指明热标触发表现的内容，如用 ! IMAGE 属性表示以下各元组中该属性中字符为图像对象名。属性中的元组有多个，每个元组又对应不同的内容，所以在把属性当作热源时，就要对每一个元组都能指明不同的链（转移方向）。元组改变，方向也就改变。

#### 4. 宏节点

宏节点是指链接在一起的节点群，更准确地说，一个宏节点就是超文本网络的一个有某种共同特征的子集。当超媒体信息网络十分巨大时，或者该信息网络分散在各个物理地点上时，仅通过一个层次的超媒体信息网络管理会很复杂，因此分层是简化网络拓扑结构最有效的方法。国外有人专门定义了宏文本 (macrotext) 和微文本 (microtext) 的概念，来表示不同层次的超文本。微文本又称小型超文本，它支持对节点信息的浏览；而宏文本又称大型超文本，由多个微文本（称为宏节点）组成，支持对微文本（即宏节点）的查找与索引。宏文本强调存在于许多文献之间的链，构造出文献相互间的关系，查询与检索将跨越文献进行。例如，在计算机网络上，很多超媒体的 Web 网分散在多台计算机中，这些 Web 网称为宏节

点或文献，它们之间通过跨越计算机网络的链进行链接，而多个宏节点或文献将组成宏文献。如图 6-6 所示。很显然，跨越网络的超链将需要更复杂的协议支持。

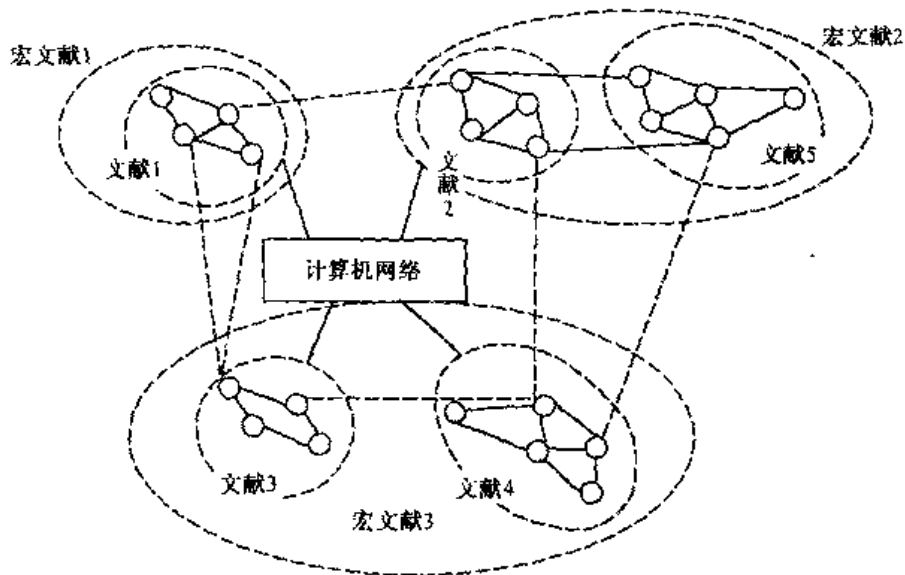


图 6-6 宏文献和宏节点示意图

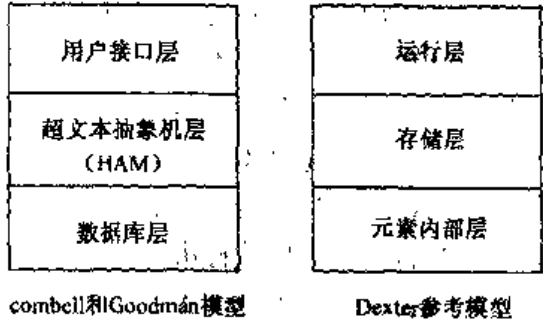
宏节点的引入虽然简化了 Web 网络结构，但却增加了管理与检索的层次。宏文本文献的查询与检索也是研究的主要问题之一，现已推出了许多模型系统，如 SMART 宏文本系统（康奈尔大学研制）、电子黄页系统（ETP），同 Guide 超文本系统相结合的宏文本系统 INDEX 系统等等。事实上，宏文本与微文本之间的界限是十分模糊的，但在应用上却可以令人一目了然，符合常规的信息存取习惯。

### 6.2.2 超媒体和超文本系统结构模型

超文本与超媒体系统就是完成生成并只使用前述超文本或超媒体的软硬件的总称，从理论上讲可将其划分为三个层次：表现层——用户接口；超文本抽象机层——节点和链；数据库层——存储、共享数据和网络访问。这是 Combell 和 Goodman 提出的一个比较标准的超文本系统结构模型。但由于国际标准尚未形成，所以实际上现有的超文本与超媒体系统大都没有完全遵循这个系统结构模型。或多或少地混入了其他特征。Dexter 小组也提出了一种 Dexter 参考模型。与上述模型比较起来，Dexter 参考模型除了术语不同并且更加明确了层次之间的接口之外，两个模型还是基本相似的。图 6-7 给出了两个模型的层次图。

#### 1. HAM 模型

1988 年，Campbell 和 Goodman 提出 HAM（Hypertext Abstract Machine，超文本抽象机）模型。HAM 模型把超文本系统划分为 3 个层次：用户界面层、HAM 层、数据库层。HAM 模型如图 6-8 所示。



combell和Goodman模型 Dexter参考模型

图 6-7 超文本与超媒体系统的两个模型

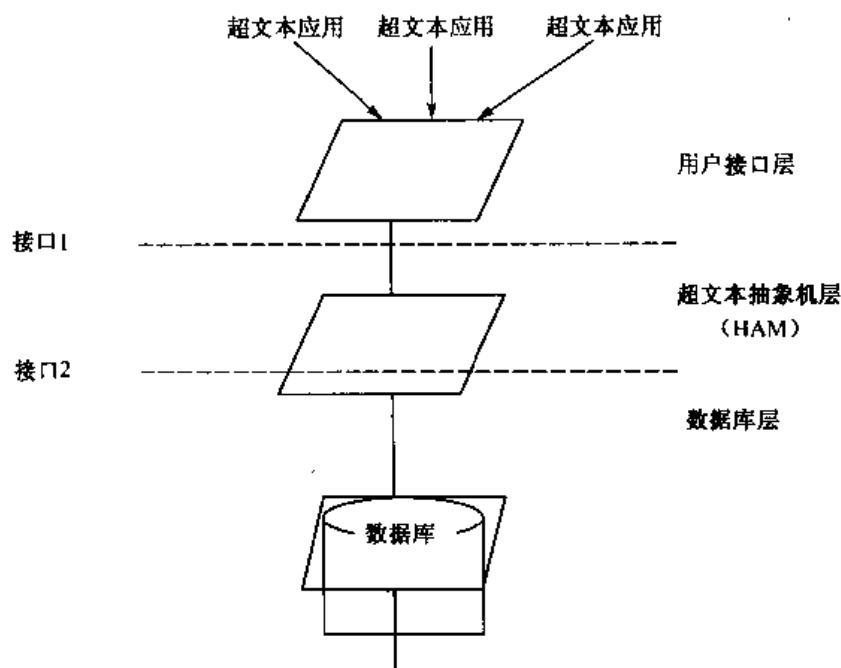


图 6-8 HAM 模型

(1) 数据库层。数据库层是模型中的最底层，比普通的数据库管理系统更为简单，用于处理所有信息存储中的传统问题。首先它要保证信息的存取操作对于高层的超文本抽象机，来说是透明的，即无论高层访问的信息是存储在本地在远地，是存储在一台计算机中还是存储在多台计算机中，数据库层都能保证正确存取。不仅如此，数据库层还要处理其他传统的数据库问题，如多用户并发访问信息的安全性、版本维护以及响应速度等问题。另外，就数据库层而言，超文本的节点和链都是没有什么特殊含义的数据对象，它们各自占据若干位的存储空间，构成在同一时间只有一个用户可修改的单元。因此，对于数据库层来说，了解比它的数据对象的信息更多的信息，是很有用的。增加对节点和链的索引和查询信息，是为了更有效地管理数据空间并提高响应速度。因此在超文本数据库层的设计中，实际上用了大量传统数据库的思想和方法。

(2) 超文本抽象机层。超文本抽象机层简称 HAM，介于数据库及用户接口层之间，这一层决定了超文本系统节点和链的基本特点，记录了节点之间链的关系，并保存了有关节点和链的结构信息。在这一层中可以了解到每个相关联的属性，如每个节点都有“主”属性（如用户修改权限、版本号或关键词等）。链则可定义其类型（如基本链，交叉索引链等）、访问权限、版本、条件等等。

另外，虽然超文本与超媒体系统还没有统一的标准，但最终标准化工作要求超文本系统之间必须具有相互传送或接收信息的能力，因此必须提供信息转换格式。而 HAM 层就是实现超文本输入输出格式标准化转换的最佳层次。因为数据库层在存储格式上依赖于不同的机器。用户接口层各个超文本系统又各具风格，不尽相同，因此难以统一。超文本的格式转换比在节点中简单地转换混合数据要困难得多，尽管它也存在非 ASCII 码信息的非标准数据格式问题（如图形、视频等），但问题是超文本的转换不仅要转换节点中的多媒体混合数据，还要求传送信息之间的链接关系，它可能传送了基本链（如  $A \rightarrow B$ ）的信息，但丢失了其他



的链接信息。例如有些超文本系统如 Intermedia, 就有这样的链, 它可以指向在目的节点中一个特殊字符的字符串, 而有些超文本系统如 Hyperties 的指针则指向整个目的节点, 那么从 Intermedia 到 Hyperties 超文本转换或传送就要丢失一些重要的链接信息。

(3) 用户接口层。用户接口层处理 HAM 中的信息表现, 包括诸如什么命令对用户有效, 如何显示节点和链, 是否要包括总体图解及多媒体信息的表现组织等。我们可以假设超文本系统的 HAM 层定义了许多种类的节点和链, 但在用户接口层它可以根据用户的权限规定哪些节点和链是可见的, 哪些是不可见的。例如在辅助教学应用中要求学生回答一个问题, 触发答案的链及答案节点对于学生来讲就是不可见的, 但对教师却应是可见的。另外导航工具也是用户接口层的一个组成部分, 它是用于浏览、查询节点、防止用户迷路的交互工具。它可以用图形化的方式表示出一个超文本或超媒体网络的结构图, 与在数据库层中存储的节点和链一一对应, 这种导航工具被称之为导航图(或浏览器)。总之, 用户接口层是超文本和超媒体系统人机交互的窗口, 随着多媒体技术的发展, 用户接口层也会越来越接近于人的交互方式。

## 2. Dexter 模型

1980 年, 由 J.Leggett 和 J.Walker 发起组织了一个研究超文本模型的团体, 以后逐渐发展成了一个超媒体参考模型, 并以当时讨论地旅馆的名字 Dexter 命名, 简称 Dexter 模型。这个模型的目标是为开发分布信息之间的交互操作和信息共享提供一种标准或参考规范。

Dexter 模型如图 6-9 所示分为三层, 即存储层(storage layer)、运行层(run-time layer)和元素内部层(within-component layer)。各层之间通过两个接口: 锚定接口(anchoring)和表现规范(presentation specification)相互连接。

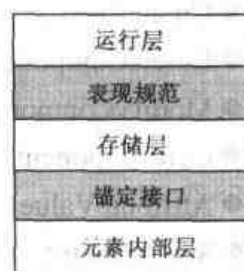


图 6-9 Dexter 模型的层次

(1) 存储层。Dexter 模型的关键是存储层, 因为存储层描述了超文本系统最基本的也是最重要的元素之间的网状关系。实际上, 存储层定义了由元素组成的数据模型。这里, 元素是对超媒体系统中基本组成单元的抽象描述, 也就是前述的节点和链等。在各个系统对元素可以采用不同的名称, 如在 NoteCards 和 HyperCard 系统中称节点元素为卡, 在 KMS 系统中称为帧, 在 Intermedia 系统中称为文献。在节点元素中的信息可由各种媒体组成。存储层的描述着重于定义元素间的连接关系, 而不涉及元素的内部结构。

在存储层的描述中, 超媒体是由一个有限元素组成的集合和两个函数组成:

$$\text{Hypermedia} = (E_1, E_2, \dots, E_n, F_1, F_2)$$

其中  $E_1, E_2, \dots, E_n$  表示有限个元素,  $F_1$  和  $F_2$  是两个用于检索定位的函数, 一个称为分解函数(resolver), 另一个称为访问函数(accessor)。

在存储层中最基本的单元是元素, 一个元素可以是原子单元、链, 或者是由原子单元和链组成的复合单元, 更为复杂的是由复合单元和链组成的复合单元。原子单元是存储层中最简单的单元, 原子单元的内部结构在元素内部层中描述。

链是用于表示元素与元素之间关系的一种实体。一般情况下链是由两个或多个元素“节点”组成的点序列。在 Dexter 模型中, 链的形式多种多样, 最常见的链是两个节点组成的, 一个称为源节点, 一个称为宿节点。由于链是一种元素, 因此一个链也可以作为另外其

他链的“节点”。Dexter 模型还支持多头链 (Multiheaded link)，主要用于从一个元素同时检索到多个元素的情况。它也支持少于两个节点的链，称为悬挂链 (dangling link)。由于 Dexter 模型支持悬挂链，因此在超媒体系统中不再需要定义起始链和终结链。

在 Dexter 模型中，每个元素都有一个惟一的标识符，称为 UID。从整个超媒体系统的全局来看，每个元素的 UID 都是不同的。访问函数的功能就是当用户指定某个元素的 UID 时，能够在超媒体系统中成功地定位找到该元素。由于 Dexter 模型中元素的检索定位完全依赖于链，因此有时仅指定某个元素的 UID 并不能马上找到该元素。这时就需要分解函数起作用。分解函数的功能是当用户指定某个元素的 UID 而不能直接找到该元素时，需要将目标元素的 UID 分解为由一个或多个中间 UID 组成的集合，这样访问函数就可根据中间 UID 集合中的成员找到目标 UID 指定的元素。

在存储层中还定义了一个操作集合，它由多个函数组成。这些函数的功能主要是实时地对超媒体系统进行访问和修改。操作集合中的操作包括在超媒体系统中增加一个元素、删除一个元素、修改一个元素的内容及其附加信息等。主要操作函数如下：

- CreateComponent。创建一个新的元素并把它加到超媒体系统中。
- CreateAtomicComponent。创建一个仅由原子单元组成的元素。
- CreateLinkComponent。创建一个链。
- DeleteComponent。删除一个元素。
- ModifyComponent。修改一个元素。
- GetComponent。当给定元素 UID 时，返回该 UID 对应的元素。
- AttributeValue。当给定元素 UID 和属性时，返回该元素的属性值。
- SetAttributeValue。给定元素 UID、属性和属性值时，将该属性设定为该属性值。
- AllAttribute。返回所有元素的属性集合。
- LinkToAnchor。给定元素 UID 和内部锚号时，返回所有指向该锚的链。
- LinkTo。给定元素 UID，返回指向该元素的所有链或由链组成的路径。

(2) 元素内部层。元素内部层定义了各个元素内部的不同内容和结构。元素内部层并不是 Dexter 模型的核心，但是必不可少的组成部分。在一个元素中，其内容和结构是没有限制的，从内容来说，可以是任何媒体的任何可用数据模型；从结构上来说，元素可由简单结构和复杂结构组成。简单结构就是每个元素的内部仅由同一种数据媒体组成，而复杂结构的元素内部又由各个子元素组成，而子元素的结构又与元素相同。这种嵌套结构的元素定义为描述复杂的混合类型的元素提供了灵活性和多样性。在 Dexter 模型中，元素内部层是开放的，也就是说，Dexter 模型对元素内部的实现不作硬性规定，可由应用程序根据实际情况作出灵活处理。

(3) 运行层。在存储层和元素内部层定义的数据及其时序和链接关系对用户来说是不可见的。运行层则为用户提供了一种可视可听的工具。它可以直接访问和操作在存储层和元素内部层定义的网状数据模型。

在运行层中最基本的概念是元素例示 (instantiation of a component)。例示的过程实际上就是将元素播放给用户。在实现过程中，首先将元素的内容复制到缓冲区，用户可编辑或浏览缓冲区内的内容。当缓冲区的内容被改变后，运行层会定时将缓冲区的内容备份到存储层中。

(4) 定位机制。定位机制通过锚定接口完成。由于在 Dexter 模型中，描述元素间链接关系的存储层和描述元素内部结构的元素内部层是各自独立的，在检索定位的过程中就需要

一个接口来维护从存储层到元素内部层、元素内部层到存储层的检索定位过程。在 Dexter 模型中, 介于存储层与元素内部层之间的接口称为锚定接口, 由它来完成定位工作。

锚定接口的基本组成部分是锚(anchor)。锚由两部分组成: 锚号(anchor id)和锚值(anchor value)。锚号是每个锚的标识符, 锚值用来指定元素内部的位置和子结构。从存储层来看锚值是没有意义的, 只有元素内部层的应用程序才能解释锚值。锚号不同于元素的 UID, 元素的 UID 从整个超媒体系统来看都是不同的, 而锚号只是在各个元素内部编号不同。因此, 从整个超媒体系统全局来指定一个锚则必须使用 (UID, 锚号) 才行。

在 Dexter 模型中, 锚号是一个相对固定的值, 而锚值则是一个经常变化的值。由于超媒体系统的元素内部结构在运行过程中可能会改变, 因此元素内部层的应用程序必须实时调整锚值的变化, 保持它原有的指向。锚与链的不同之处在于链仅指向元素, 而锚则可指向元素内部的具体内容。

(5) 表现规范。在 Dexter 模型中介于运行层与存储层之间的接口称为表现规范。表现规范规定了同一数据呈现给用户的不同表现性质。比如在一个教学系统中, 播放给教师看的习题可以有答案, 而播放给学生看的同一习题就没有答案。对操作也是如此, 给用户播放的不允许编辑, 而播放给系统设计者的就允许对其进行编辑。

Dexter 模型的提出, 为超媒体的设计起了重要的指导作用。相当多的系统虽采用了这个模型, 但也很不一致。许多新的模型还在进一步的研究之中。

## 6.3 超文本和超媒体的文献模型

### 6.3.1 概述

文献是文章或文本的组合, 它比一般文章和文本带有更多的存储、保留的意味, 一旦定形后静态性较强。超文本的文献模型侧重于超文本的基本特征和一般的层次性结构的描述。文献结构可分为主结构和次结构。主结构是占优势的层次结构, 它定义目标如何结合成更高层次的目标; 次结构表达目标间的附加关系。超文本系统强调次结构, 但没有很好的定义主结构, 用户会产生混乱。

#### 1. 文献的一般结构

文献的结构包括内容组织和版面安排两个方面。内容组织指作者在不考虑版面的情况下如何组织和构造文献的信息内容; 版面安排是相对于内容的表现形式来说的, 即文献的各部分内容如何安排在每一页面(屏幕)上。

如把一本书当作文献来看, 其内容组织主层次结构是: 文献(内容)→章→节→小节→段落(内容实体)。内容实体可以是一段文字或是一个图表。相应地, 也可以把版面安排层次化, 以便和内容组织对应, 可以分成: 文献(版面)→页/整屏→框架/窗口→块/子窗口(内容实体)。

#### 2. 文献模型的基本任务

多媒体文献模型与普通文献相比在内容实体的媒体属性方面更加多样化, 尤其是时基媒体的引入, 不仅要考虑文献的版面安排, 还要考虑时间安排。因此, 多媒体文献模型的基本任务是:

- (1) 能够表示多媒体文献的内容层次性。
- (2) 能够表示多媒体文献的版面布局。
- (3) 能够表示多媒体文献的时间布局。
- (4) 能够将内容与布局对应起来。

下面介绍的 ODA 模型仅有版面布局，而国际标准 HyTime 还考虑了时间布局。

### 6.3.2 ODA 模型

ODA (the Office/Open Document Architecture) 是 ISO 在 1988 年公布的一个标准化文献模型 (ISO 8613: 1988)，它为辅助办公文献的表示和交互而设计。它提供了文献的静态描述，还提供了与其他文献格式的接口。

#### 1. ODA 文献结构

ODA 文献结构是层次的和面向对象的。一个 ODA 文献由两对结构来描述，即一般结构和具体结构，逻辑结构和布局结构。前者体现面向对象性质，后者体现内容与表现的关系。

(1) 逻辑结构和布局结构。文献的内容层次性用逻辑结构描述，它首先按文献内容划分成逻辑目标，这些目标对作者或读者意味着某些事情。一个逻辑目标可以是一个一般项，如书中的一节、标题、段落等。它也可以是特殊项，如电话号码、价格或者产品的清单。只有最低层的目标才有内容。

文献的版面安排用布局结构描述。它按内容划分为页集、页和页中方框区域，其中定义有嵌套区域的方框区域称为框架 (frame)，最低层的区域称为块。块是惟一有内容与之关联的区域。

我们以教科书的一章为例介绍这两种结构的对应关系。章从逻辑上可分为节、小节、段、内容，从布局上分为页、框架、块 (实体)。两种结构相互依赖、相互对应，在最下层的“内容实体”一层达到统一。两种结构的对应关系如图 6-10 所示。

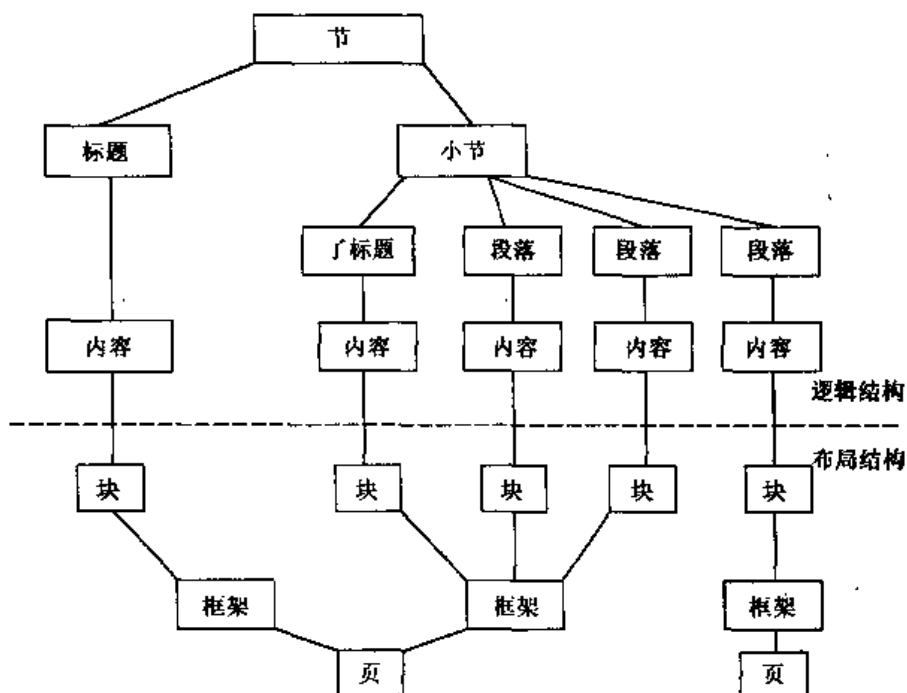


图 6-10 逻辑结构和布局结构的对应关系

(2) 一般结构和具体结构。每个文献都有具体的 (specific) 逻辑结构和布局结构, 而具体结构的建立是由相应的一般 (generic) 结构控制的。一般结构是一系列关于对象的定义 (对象分为逻辑对象集合和布局对象集合)。

每个非页节点的对象定义都有一个属性“从属产生器” (generator for subordinates), 用来说明对象如何由其子对象构成。从属产生器属性有:

- 可选的 (OPT) 0 或 1 次事件。
- 要求的 (REQ) 仅 1 次事件。
- 重复 (REP) 1 次或多次事件。
- 可选并且重复 (OPT REP) 0、1 或多次事件。
- 一种顺序 (SEQ) 以固定顺序出现。
- 一种聚集 (AGG) 以任意次序出现。
- 一种选择 (CHO) 仅其中一个被选中。

图 6-11 表示期刊中某篇论文的一般逻辑结构。它指出论文必有标题, 跟着是必有的作者名, 接着是一个可选的摘要和一个或多个章节。如果存在摘要, 则它由一段或多段组成。每节由子标题开始, “REP CHO” 结构指出子标题后跟一系列段落或列表, 它们是以任意顺序出现的。列表包含一个或多个项。

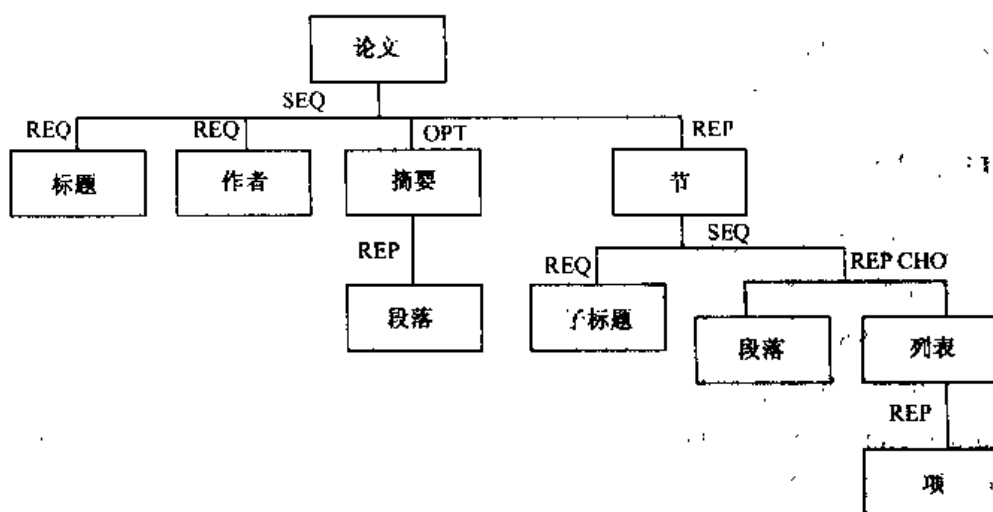


图 6-11 一般逻辑结构

相应的一般布局结构可以对论文的第一页定义一个风格, 对其余的页定义另一个不同的风格, 图 6-12 就表示了这种结构。标题页包括一个标题框架 (header frame) 和一个正文框架 (body frame)。标题框架是为标题、作者名和摘要设立的表示区域, 而正文框架则作为文章第一部分的开始。“继续页”包含“继续正文框架”以保存章节的其余部分。块并不包含在一般布局结构中, 但在布局过程中它被分派到页或框架中。

## 2. 布局过程

ODA 的布局过程确切地决定文档中的每一项被放置的位置。它使用特定的逻辑结构、一般结构、内容体系以建立特定的布局结构。它工作在两个层次上:

(1) 内容布局处理内容部分, 并将它们安排到块中。这个阶段依赖于涉及到的内容结构和称为表达风格的属性集。

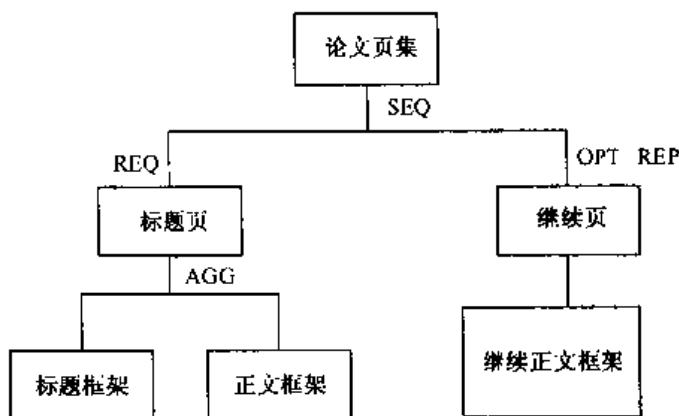


图 6-12 一般布局结构

(2) 文献布局将块安排到框架或页中。这个阶段依赖于称为布局风格的属性集。

内容布局处理字符集和项在块内的合理定位，更高一级的文献布局过程决定如何将块置入页或框架中。

布局目标类通常用来指定文档的主逻辑划分和特殊页或页集的对应关系。在一个布局目标类中，布局类别和允许类别属性能被用来把逻辑目标引导到不同的框架中。如果对一个页逻辑目标给定一个布局类别名，则它只能被放置在具有相同名字的框架中，此名为允许类别名之一。当特定布局结构被创建后，它将文献和页、框架、块联系起来，图 6-10 显示了在逻辑结构和布局结构的一一对应。

### 6.3.3 HyTime 模型

HyTime 全称为“Hypermedia/Time-based Structuring Language”（时基超媒体结构化语言），它是一个标准的中性标记语言，用以表示超文本、多媒体、超媒体和时基文献的逻辑结构。HyTime 从 1986 年 6 月起由 ANSI 的一个工作组开发，后被 ISO 采纳，于 1992 年 5 月成为 ISO 国际标准，其标准号为 ISO/IEC 10744: 1992。HyTime 基于 SGML (ISO/IEC 8879: 1986, Standard Generalized Markup Language)，用 HyTime 表示的文献与 ISO SGML 完全一致，HyTime 扩展了 SGML，使 SGML 更具抽象性、中立性，且增加了许多关于多媒体应用方面的考虑。

#### 1. SGML 元素

SGML 用国际标准的“标签”（tag）语法来标记一个数据合成体中各块信息的组成情况。如果“数据合成体”是一个文献，那么用 SGML 标记过的文献就是一个 SGML 文献。

(1) SGML 元素。元素是一个可标记的逻辑体，以“book”为例，视 Book 为一类元素，将它可分为若干 Chapter，Chapter 还可分为 Title 和若干 Section。Chapter，Title 和 Section 也是元素。它们都是含有一定结构的逻辑体。一个元素的标记实例见图 6-13。图 6-14 是一个 Book 类元素的实例。

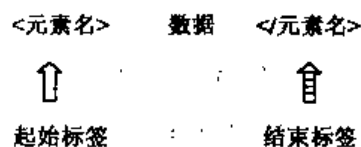


图 6-13 SGML 元素

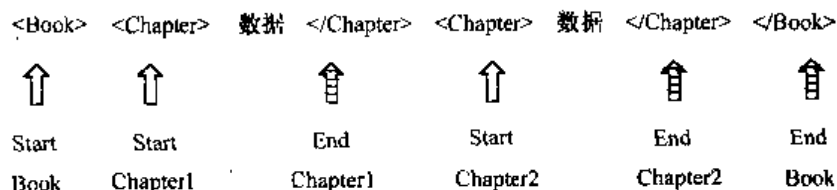


图 6-14 Book 类元素的实例

(2) SGML DTD。在 SGML 中，用 DTD (document type definition) 来定义文献 (元素) 的类型，描述其内部的一般逻辑结构。这就是元素的一般定义，像上面的例子就有如下 DTDs:

```
<! ELEMENT Book-- (chapter +) >
<! ELEMENT Chapter-- (Title, Section +) >
<! ELEMENT TitleSection--CDATA) >
其中+表示一个或多个。
```

(3) SGML 属性。SGML 用“属性”的方法来表示对某一个元素的必要的非结构化的信息。属性由“属性名”和“属性值”组成。属性名及其值包含在起始标签里面，是标签的一部分而非数据的一部分，如下所示:

```
<Book author= "JOGN" >...(Chapters)...</Book>
```

相应地，DTD 变为:

```
<! ELEMENT Book--(Chapter +)>
<! ATTLIST Book author CDATA #REQUIRED>
```

(4) SGML 惟一标识符。有两种特殊的属性值: ID 和 IDREF。如果一个元素有一个 ID 类型的属性，那么其值必须是该元素的惟一名字。如果因为某种原因，另外一个元素 A 要引用一个有惟一名字的元素 B，那么 A 的属性 IDREF 的值就是 B 的惟一名字。基于 SGML 的超媒体系统就用这种机制来表示文献内部的超链。

(5) SGML 实体 (entity)。如前所述，有两种 SGML 支持的结构: 元素的层次结构以及隐藏于 ID 和 IDREF 机制中的有向图式的任意结构，第 3 种结构是实体结构。在 SGML 中，一个实体是任意数据资源，可以是文件、硬件子系统、存储缓冲区等等。如果一个完整的 SGML 文献包含在一个单一的文件中，那么这个单一的文件是该 SGML 系统需要考虑的惟一实体。在一个较复杂的 SGML 文献中，“实体引用”就会发生。

实体定义形式为: <! ENTITY...>。

如<! ENTITY "Myentity" SYSTEM "Usr/Local/text/myentity" >，即如果在文献中发现 "&Myentity;" 字样，就会启动 SGML 系统来引用该实体。

## 2. HyTime/SGML: 元 DTD

SGML 是 HyTime 的基础。SGML 提供了作为逻辑“元素”和物理“实体”的语法表示以及将信息组织或层次结构的“文献”的语法。HyTime 利用称为“SGML 结构形式”的形式，提高了 SGML 的抽象性和中立性，可以用来更好地表示多媒体文献系统的特性。

(1) 元 DTD (meta DTD)。如果两个不同的用户分别开发的 DTDs 互不相同，后来发

现自己的信息需要与另一个交互，于是需要将他们的不同的 DTD 一致起来，相应地重新整理各自的信息。于是有必要提出更高层次上的抽象。HyTime 提出了比 SGML DTD 更抽象的元 DTD。

元 DTD 形式并不描述一个 DTD，但它在人们建立实际 DTD 时给予严格的格式指导。遵守元 DTD 的好处是软件重用性提高，而处理费用却相应减少。

元 DTD 由一系列组织和包装好了的“SGML 结构形式”构成。SGML DTDs 中的一个 `<!ELEMENT...` 定义对应文献的类元素，正如元 DTD 中的一个 SGML 结构形式对应 DTDs 中的一类 `<!ELEMENT...` 定义。即，如果称一个 `<!ELEMENT...` 定义是一个元-元素的话，那么一个结构形式则是一个元-元-元素，因为 DTD 中的一个 `<!ELEMENT...` 定义表示的是实际文献中一类元素的内部结构和属性必须遵守的规则，而元 DTD 的一个结构形式表示的是实际 DTDs 中一类 `<!ELEMENT...` 定义的内部结构和属性必须遵守的规则。所以元 DTD 之于 DTD 正如 DTD 之于文献。元 DTD 是最抽象的结构，是最中立（独立）于具体应用的。HyTime 其实就是一个元 DTD 系统。

(2) HyTime 的 3 种基本逻辑结构。SGML 有 3 种结构：物理的实体结构、逻辑的元素结构及类似指针的 ID-IDREF 结构。HyTime 取其优点，加以提高。HyTime 支持 3 种基本的逻辑结构：

1) SGML 层次性元素结构。这种结构可看作是一棵树，树的每个节点是一个元素。在 SGML 语法中，父元素包括其所有的孩子元素以及孩子元素的孩子元素，如此类推。

2) HyTime 超链结构。这种结构可以在节点之间建立非层次性关系。超链结构类似于 SGML 的以 ID-IDREF 表示的任意有向图结构，但它又大有提高。如链尾不必是一个元素或实体；一个链可以有若干链尾，分别通向各自的语义目的地；可以表示复杂的遍历语义等。

3) HyTime 调度结构。HyTime 调度结构可以不重复某一个数据，而表示出若干与该数据相关联的“时间表”。这使人联想到数学里的坐标系，如最常见的三维空间  $(x, y, z)$ ，该空间中任一点都对应着 3 个坐标轴上不同的值。HyTime 的有限坐标空间 (finite coordinate spaces: FCSs) 便如是产生。FCSs 坐标空间的各个坐标轴就是前面说的所谓“时间表”。

“时间表”的含义并非一定就是指时间，它还可以是其他任何与数据相关联的约束或说明范畴。图 6-15a, 6-15b, 6-15c 简单显示了上面 3 种结构。

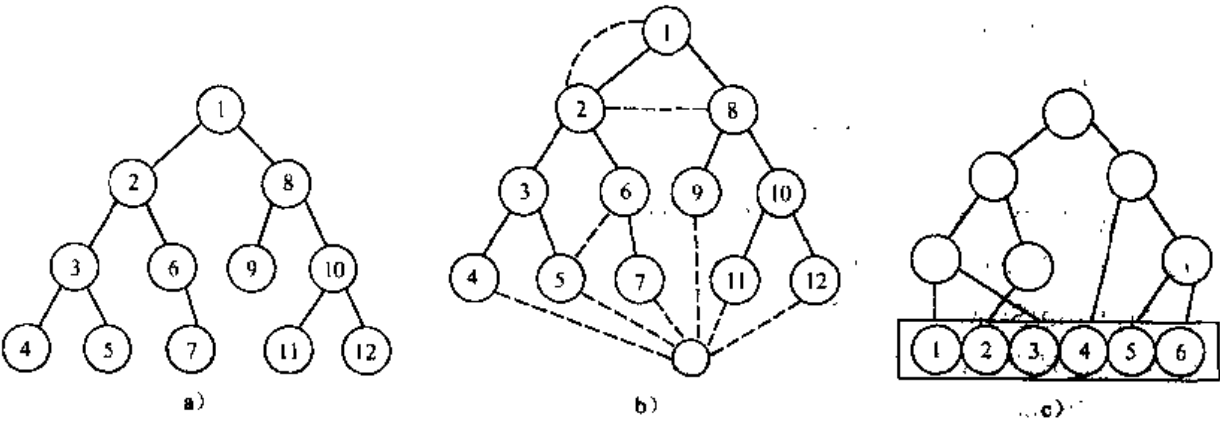


图 6-15 HyTime 的 3 种结构

a) SGML 层次结构 b) HyTime 超链结构 c) HyTime 调度结构



## 6.4 超文本与超媒体存在的问题及发展前景

### 6.4.1 超文本与超媒体存在的问题

超文本与超媒体是一项正在发展中的技术，它虽然有许多独特的优点，但也存在许多不够完善的方面。这些问题也处于积极的研究中，具体表现在：

(1) 信息组织。超文本的信息是以节点作为单位。如何把一个复杂的信息系统划分成信息块是一个较困难的问题。例如一篇文章，一个主题，又可能分成几个观点，而不同主题的观点又相互联系，而为这些联系分割开来，就会破坏文章的本身表达的思想。这样节点的组织 and 安排就可能要反复调整和组织。

而且有些信息根本不适合组织成超文本的形式，超文本系统也没有提供工具来帮助用户完成信息的划分工作。

(2) 智能化。虽然大多数超文本系统提供了许多帮助用户阅读的辅助信息和直观表示，但因超文本系统的控制权完全交给了用户，当用户接触一个不熟悉的题目时，可能会在网络中迷失方向。要彻底解决这一问题，还需要研究更有效的方法，这实际上是要超文本系统具有某种智能性，而不是只能被动地沿链跳转。超文本在结构上与人工智能有着相似之处，使它们有机的结合将成为超文本与超媒体系统的必然趋势。

(3) 数据转换。超文本系统数据的组织与现有的各种数据库文件系统的格式完全不一样。引入超文本系统后，如何为传统的数据库数据转换到超文本中也是一个问题。

(4) 兼容性。目前的超文本系统大都是根据用户的要求分别设计的，它们之间没有考虑到兼容性问题，也没有统一的标准可循，所以要尽快制定标准并加强对版本的控制。标准化是超文本系统的一个重要问题，没有标准化，各个超文本系统之间就无法沟通，信息就不能共享。

(5) 扩充性。现有的超文本系统有待于提高检索和查询速度，增强信息管理结构和组织的灵活性，以便提供方便的系统扩充手段。

(6) 媒体间协调性。超文本向超媒体的发展也带来了一系列需要深入研究的问题，如多媒体数据如何组织，各种媒体间如何协调，节点和链如何表示；对音频和视频这一类与时间有密切关系的媒体引入到超文本中，对系统的体系结构将产生什么样的影响，当各种媒体数据作为节点和链的内容时，媒体信息时间和空间的划分，内容之间的合理组织都是在多媒体数据模型建立时要认真解决的问题。

### 6.4.2 超文本与超媒体发展的前景

(1) 由超文本向超媒体发展。从超文本到超媒体是技术发展的进步，也是技术发展的必然性。超文本向超媒体的转变不仅是将文本媒体扩展到其他媒体，而且还要能使系统自动地判断媒体类型，并执行对应的操作。对图像的热区，视频的热点等都能引起类似于热字的反应，多媒体的表现及基本内容的检索等。超文本向超媒体的转变，大大地增强了功能和性能，也增加了系统实现的难度。

(2) 由超媒体向智能超媒体发展。在超媒体技术的研究中，有人提出智能超媒体或专

家超媒体 (Expertext)。这种超媒体打破了常规超媒体文献内部和它们之间严格的链的限制, 在超媒体的链和节点中嵌入知识或规则, 允许链进行计算和推理, 使得多媒体信息的表现具有智能化。

(3) 由超媒体向协作超媒体发展。超媒体建立了信息之间的链接关系, 那么也可用超媒体技术建立人与人之间的链接关系, 这就是协作超媒体技术。超媒体节点与链的概念使之成为支持协同性工作的自然工具。协同工作使得多个用户可以同一组超媒体数据上共同进行操作。这样未来的电子邮政、公共提示板等都可能应用到超媒体系统中。

## 6.5 小结

本章讨论了超文本与超媒体的概念及其的特点和特性, 以及超文本与超媒体的体系结构, 分为三个层次: 表现层、超文本抽象机层、数据库层。以及超文本与超媒体组成的要素等。超文本与超媒体的应用和超文本与超媒体存在的问题及发展前景。

超文本与超媒体技术是通过节点和链将信息构成一个网状的互联结构。这是一种非常实用的技术, 特别是对多媒体的信息管理, 更突出它的特色。

超文本和超媒体也有其不完善的地方, 但随着多媒体技术的发展, 超文本和超媒体将向标准化、智能化发展, 将不断完善其体系结构。

## 6.6 习题

1. 在超文本和超媒体中不同信息块之间的连接是通过\_\_\_\_\_连接。

- |        |        |
|--------|--------|
| (A) 节点 | (B) 字节 |
| (C) 链  | (D) 字  |

2. 超文本的三个基本要素是\_\_\_\_\_。

- |                 |                 |
|-----------------|-----------------|
| (1) 节点          | (2) 链           |
| (3) 网络          | (4) 多媒体信息       |
| (A) (1) (2) (4) | (B) (2) (3) (4) |
| (C) (1) (3) (4) | (D) (1) (2) (3) |

3. 超文本和超媒体体系结构的三层模型是一年提出的?

- |          |          |
|----------|----------|
| (A) 1985 | (B) 1988 |
| (C) 1989 | (D) 1990 |

4. 下列的叙述哪些是正确的?

- (1) 节点在超文本中是信息的基本单元。
- (2) 节点的内容可以是文本、图形、图像、动画、视频和音频。
- (3) 节点是信息块之间连接的桥梁。
- (4) 节点在超文本中必须经过严格的定义。

- |                 |             |
|-----------------|-------------|
| (A) (1) (3) (4) | (B) (1) (2) |
| (C) (3) (4)     | (D) 全部      |

5. 下列的叙述哪些是错误的?

(1) 链的结构分为三部分：链源、链缩及链的属性。

(2) 链是连接节点的桥梁。

(3) 链在超文本中必须经过严格的定义。

(4) 链在超文本和超媒体中是统一的。

(A) (1) (2)

(B) (1) (3)

(C) (3) (4)

(D) 全部

6. 超文本和超媒体系统中的数据库与传统的数据库有什么不同？

7. 超文本和超媒体的组成要素与操作工具有哪些？

## 6.7 参考答案

1. (C)

2. (D)

3. (B)

4. (B)

5. (C)

6. 答：超文本和超媒体系统中的数据库层，是模型中的最低层，比普通的数据库管理系统更为简单，用于处理所有信息存储中的传统问题。如存储分配管理、缓冲区调度、存储控制等等，其基本功能是对节点和链等的基本信息进行存储，管理和访问，并保证这些操作对于高层的超文本抽象机层来说是透明的，即无论高层访问的信息是存储在本地或远地，是存储在一台计算机在还是存储中在多台计算机中，数据库层都能保证正确的存储。

超文本和超媒体系统中的数据库，由于具有多媒体数据所以信息量很大，因此用到大容量的存储技术，如大容量的磁盘、光盘等。而传统的数据库信息量没有那么大，而且信息量比较单一，一般都是文档或数据等。但是在超文本和超媒体的数据库层的设计中也用到了大量的传统数据库的思想方法。

7. 答：超文本和超媒体的组成要素是由节点、链、和网络等组成。其中节点可分为表现型记录各种媒体信息，另一种是组织型用于组织并记录节点间的联结关系。

节点可归纳为以下基本类型：文本节点、图形节点、图像节点、音频节点、视频节点、混合媒体节点、按钮节点、组织型节点和推理型节点。

链：链由三部分组成，即链源、链缩和链的属性。而链的种类可分为：基本结构链、（基本链、交叉检索链、节点内注释链）、推理链、隐形链等。

网络：即超文本和超媒体数据库由声、文、图各类节点和链组成网络。

超文本和超媒体的操作工具主要有：编辑器、编译器、阅读器、导航工具等，而导航工具又可分为：导航图、查询系统、线索、遍历、书签等。

## 第7章 多媒体著作工具

### 本章要点

- 多媒体著作工具是一种高级的软件程序或命令集合。这些命令可以支持各式各样的硬件设备与文件格式。将图形、文本、动画、声音等不同类型的信息等组合在一起，并进一步提供一个导向结构，使多媒体系统的设计者具有一个良好的集成环境，帮助设计者将各种内容与各种不同的功能结合在一起，组成一个结构完整的系统。根据多媒体著作工具的创作方法和特点的不同，可将其划分为四类。
- Authorware 是美国 Macromedia 公司于 1991 年 10 月推出的多媒体著作工具，它可以在 Windows 9x、Windows Me、Windows 2000 和 Windows NT 等环境下运行的多媒体制作软件。

开发多媒体应用程序，除需要较强的编程能力外，还需要有专业性很强的多媒体信息处理的知识，因此，在 20 世纪 90 年代初，不少人感叹地说：“多媒体是个好东西，但它仅仅是专家的乐园”。应用程序的开发成为多媒体技术应用推广的一个瓶颈。为了解决这一问题，国内外许多厂商和研究机构相继研制了一批称作“著作工具（Authoring Tool）”的一类多媒体软件。著作工具试图将开发过程中的编程工作简化，一个没有编程经验的人也可以利用著作工具制作一个具有菜单，能显示图像、文本和动画的多媒体应用程序。

多媒体著作工具是电子出版物、多媒体应用系统开发的基础工具。它提供组织和编辑多媒体项目各种成分所需要的重要框架，包括图形、图像、声音、动画和视频剪辑。

著作工具的用途是设立交互性和用户界面、在屏幕上演示制作的项目以及将各种多媒体成分集成完整而具有内在联系的项目。

全球第一个流行的多媒体著作工具为 1987 年提出的 macintosh 系列超卡 Hypercard，Hypercard 解决了人机界面设计，使每一项多媒体素材成为一个对象（object），用户可以轻易的产生、更改、变化各个图像，而不必花费很多精力去想如何写程序，从而专心编排多媒体素材，完成多元化，多样化的节目。之后，世界很多厂商设计、生产了多种多媒体著作工具，如 Authorware、ToolBook、Director 等。

### 7.1 多媒体著作工具

#### 7.1.1 多媒体著作工具的分类

多媒体著作工具是一种高级的软件程序或命令集合。这些命令可以支持各式各样的硬件设备与文件格式。将图形、文本、动画、声音等不同类型的信息等组合在一起，并进一步提供一个导向结构，使多媒体系统的设计者具有一个良好的集成环境，帮助设计者可以将各种

内容与各种不同的功能结合在一起，组成一个结构完整的系统。

因此，有了这种著作工具，即使没有计算机专业知识的使用者，也能很直观地编排出多媒体作品，这样就可以使得多媒体应用软件的开发人员把更多的精力集中放在主题的表达、交互特性的设计和用户界面的设计等方面。多媒体著作工具一般都提供了不同的应用开发环境，并具有各自的功能特点，使用于不同的应用范围。根据多媒体著作工具的创作方法和特点的不同，可将其划分为如下几类：

- 基于时间的著作工具。
- 基于图标或流程线的著作工具。
- 基于卡片或页面的著作工具。
- 以编程语言为基础的著作工具。
- 其他：如综合类的多媒体节目编制系统。

#### 1. 以图标为基础的多媒体著作工具

在这种著作工具中，数据是以对象或时间的顺序来组织的，并且以流程图为主干，将各种图表、声音、控制按钮等一个个接在流程图中，形成完整的系统。

这类著作工具有 Icon Author 和 Authorware 等。它们一般只完成多媒体素材的集成及组织工作，在设计之初必须先用其它工具性软件来制作各种媒体素材，然后在此系统中建立一个流程图，在流程图当中可以包括起始事件、分支、处理及结束等各种图标。设计者可以依照流程图将适当的对象从图标库 (Icon Library) 中拖拽至工作区内。这些图标可以包括菜单条的选项、图形、图像、声音及运算等。这个流程图也是事件安排的次序，即表示了整个应用系统的逻辑关系图。

这类著作工具的最大特点是无需编程，采用图形化的过程控制，有多种交互方式和丰富的系统变量，可以实现各种特殊视听效果。优点是调试方便，在复杂的航行中，这个流程图对开发过程特别有用。缺点是当多媒体应用软件制作得很大时，图标及与分支很多。

#### 2. 以时间为基础的多媒体著作工具

以时间为基础的多媒体著作工具所制作出来的节目最像电影或卡通片。它们以可视的时间轴来决定事件的顺序和对象显示上演的时段。在这种著作工具中，数据或事件是以一个时间顺序来组织的。它的基本设计思想如同我们日常生活中安排约会那样，用时间线的方式表达各种媒体元素在时间线上的相对关系，把抽象的时间观念予以可视化。这种工具对媒体元素之间的同步比较容易控制，较适合于制作交互特性要求不高，仅需循环播放的多媒体节目。这类多媒体著作工具使用起来如同电影剪辑，可以精确地控制在什么时间播放什么镜头，至少可以精确到每一帧，这类多媒体著作工具的典型产品有 Director 和 Action 等。

这种控制方式的优点是操作简便，形象直观，在一个时间段内，可任意调整多媒体素材的属性。其缺点是要对每一素材的呈现时间做出精确安排、调试工作量大，它适合于一项有头有尾的消息。这类多媒体著作工具特别适合于制作动画，甚至是广播级的动画片。

#### 3. 以页为基础的多媒体著作工具

在这种著作工具中，文件与数据是用如一叠卡片或书的页来组织的，它提供了一种可以将对象连接于卡片活页上面的工作环境。一张卡片或一页就是数据结构中的一个节点，它类似于数据袋里的一张卡片或教科书中的一页，只是这种卡片或页面上的数据比教科书上的页面或数据卡片上的数据类型更多样。并且，这些数据大多是用图标来表达的，使得它们很容

易理解和使用。这类多媒体著作工具是以面向对象的方式来处理多媒体元素的，这些元素用属性来定义，用剧本来规范，允许播放声音元素以及动画和数字化视频节目。在结构化的导航模型中，可以根据命令跳至所需的任何一页，形成多媒体作品。

这类多媒体著作工具的超文本功能最为突出，特别适合于制作电子图书，这类著作工具主要有 Tool Book 和 HyperCard 等。优点是便于组织和管理多媒体素材。缺点是在要处理的内容非常多时，卡片或页面数量过大，不利于维护与修改。

#### 4. 以传统程序设计语言为基础的多媒体著作工具

目前，虽然开发多媒体软件的程序设计语言不少，但在 Windows 环境下微软公司的 VB 和 Borland 公司的 Borland C++ 是多媒体软件开发人员优选的程序设计语言。使用以传统程序设计语言为基础的多媒体著作工具时需要大量编程，而且可重复性差，不便于组织和管理多媒体素材，且调用困难，还有如综合类的多媒体节目编制系统的通用性差和操作不规范等缺点。

### 7.1.2 多媒体著作工具的功能要求

由于应用目标和使用对象的不同，多媒体著作工具在功能上往往会有较大的差别。一般认为，多媒体著作工具应有以下功能。

#### 1. 良好的、面向对象的编程环境

多媒体著作工具应提供编排各种媒体数据的环境，即能对媒体元素进行基本的信息和信息流控制操作，包括条件转移、循环、数学计算、逻辑运算、数据管理和计算器管理等。多媒体著作工具还应具有将不同媒体信息编入程序的能力、时间控制能力、调试能力、动态文件输入与输出能力等。编程思路方面主要有：流程结构式，先设计流程结构图，再组织素材，如 Authorware；卡片组织式，如 ToolBook。

#### 2. 较强的多媒体数据 I/O 能力

媒体数据一般由多媒体素材编辑工具完成，由于制作过程中经常要使用原有的媒体素材或加入新的媒体，因此要求多媒体著作工具软件也应具备一定的数据输入和处理能力。另外对于参与创作的各种媒体数据，可以进行实时呈现与播放，以便对媒体数据进行检查和确认。其能力为：

(1) 能输入/输出多种图像文件：BMP、PCX、TIF、GIF、TAG 等。

(2) 能输入/输出多种动态图像及动画文件：AVS、AVI、MPG 等，同时把图像文件互换。

(3) 能输入/输出多种音频文件：wav、CD-Audio、midi 等。

(4) ODBC 数据库文件功能。

#### 3. 动画处理能力

多媒体著作工具可以通过过程控制，实现显示区的位块移动和媒体元素的移动，以制作和播放简单动画。另外多媒体著作工具还应能播放由其它动画制作软件生成动画的能力，以及通过过程控制动画中的物体的运动方向和速度，制作各种过渡特技等。如移动位图、控制动画的可见性、速度和方向；其特技功能是淡入淡出、抹去、旋转，控制透明及层次效果。

#### 4. 超级链接能力

媒体元素可分为：静态对象中的 Word, Phrase, Image 及 Screen Regime 等；基于时间的数据对象中的声音、动画、视频及 CD-Audio 等。超级链接能力是指从一个对象跳到另一个对象，程序跳转、触发、连接的能力。从一个静态对象跳到另一个静态对象，允许用户指定跳转链接的位置。允许从一个静态对象跳到另一个基于时间的数据对象或相反。

#### 5. 应用程序的连接能力

多媒体著作工具应能将外界的应用控制程序与所创作的多媒体应用系统连接。也就是从一个多媒体应用程序来激发另一个多媒体应用程序，并加载数据，然后返回运行的多媒体应用程序。多媒体应用程序能够连接（调用）另一个函数处理的程序：

(1) 可建立程序级通信：DDE (Dynamic Data Exchange)。

(2) 对象的链接和嵌入：OLE (Object Linking and Embedding)。

#### 6. 模块化和面向对象

多媒体著作工具应能让开发者编成独立片断模块化，甚至目标化，使其能“封装”和“继承”，让用户能在需要时独立使用。通常的开发平台都提供一个面向对象的编辑界面，使用时只需根据系统设计方案就可以方便地进行制作。所有的多媒体信息均可直接定义到系统中，并根据需要设置其属性。总之，应具有能形成安装文件或可执行文件的功能，在脱离开发平台后能运行。

#### 7. 良好的界面、易学易用

多媒体著作工具应具有友好的人机交互界面。屏幕呈现的信息要多而不乱，即多窗口、多进程管理。应具备必要的联机检索帮助和导航功能，甚至教学软件。使用户在上机时尽可能不借助印刷文档就可以掌握基本使用方法。此外多媒体著作工具应操作简便，易于修改，菜单与工具布局合理，有良好的技术支持。

### 7.1.3 编程与著作工具

开发多媒体应用程序有两种可选的方案：进行程序设计或使用著作工具，其选择与应用程序开发的进度和质量要求有很大关系。

著作工具试图将开发过程中的编程工作简化，用户利用它不需要复杂编程也能设计出多媒体应用程序。例如，如果你能得到一份 PowerPoint 或 Action!，你完全可以在一天之内做出一个多媒体的会议报告或产品展示。

程序设计语言，如 C、C++、BASIC、Pascal 等，常用来开发专门的应用程序。这些编程语言灵活、功能强大，能够开发范围广泛的多媒体应用程序，比如微软公司开发的 Home 系列 CD 出版物就是编程实现的。利用高级语言制作多媒体应用程序不仅需要编写许多行代码，还要求编程者具有多媒体信息处理的专业知识，对非专业技术人员来说，障碍是明显的。

一般来说，具有特色的、尤其是满足消费需求的多媒体应用程序通常采用编程语言。一些多媒体软件开发公司常常采用程序设计语言，因为它们需要设计新的具有特色的软件来吸引用户；而供训练和教育用的应用软件、演示软件和商品展示软件常采用著作工具，这是由于这些环境不存在很强的竞争压力，而用户又希望很快地完成一个应用，这正是著作工具最具诱惑力的地方。

## 1. 著作工具的优势与劣势

著作工具的主要特点首先在于它缩短了用户开发应用的时间，通常在一周甚至几天之内，用户就可以开始使用它。Microsoft PowerPoint、Macromedia Action!就带有自己的说明指导程序，能帮助用户很快地进行创作，这不过只是几小时的事。多数著作工具都提供图形界面，这些图形界面为用户设计了程序的隐含流向，并自动处理许多具体的程序指令。通常著作工具包含有许多特色功能，每个功能用户要学习大约几个小时。为了演示一个动画，程序员可能要编写许多行 C 代码，可是像 AimTech 公司的 IconAuthor 可以将演示多种动画文件的功能编进了著作工具，用户只需提供动画文件名即可。

著作工具另一个优势在于它提供了样板应用程序，这些样板对于通过简单的交互、快速拼装一个应用程序极为有用。著作工具也存在非常明显的不足。著作工具开发出的应用程序一般要比用程序语言开发的应用程序运行速度慢，因为用著作工具设计的应用程序一般要用著作工具本身或者另外一个运行版本（通常称为播放程序）才能在用户的机器上运行。例如，为运行 Action!设计的应用程序，最终用户必须有 Action!或者开发者提供的 Action!的一个运行版本 Action!Player。

著作工具比编程语言要昂贵得多。一般功能强大的著作工具零售价有数千美元，而一个很复杂的编程语言编译器仅为几百美元甚至更低。

著作工具本身是一个需要支持与维护的软件。这个软件发生问题就会在用户的应用程序中表现出来，如果用户自己不能解决，就必须重新设计应用或者等待开发者前来解决，用户比较被动。著作工具提供了一个预先确定的功能集合。尽管有些系统允许熟练用户将定制的程序包含进去，但是这样的做法也因系统而异。著作工具仅仅允许用户暂时退出应用程序去执行一个独立的软件，当外部程序运行完毕，需重新进入著作工具并返回控制。另外，增加各种功能的做法可能仅适合某些特定的应用。一个允许用户跳出其执行环境去执行一个独立程序的著作工具，可能对教育软件比较合适，但不太适合电子百科全书。实际上，采用著作工具就好比乘公共汽车，别人帮助你开车，但是你必须按照他们选定的地点上下车。著作工具的设计者要根据用户的基本需要来确定系统必须包含哪些功能，这就意味着你所需要的某些功能在一些著作工具中并不支持。

## 2. 编程的优势与劣势

程序设计语言就好像是一辆高性能的赛车，理论上，任何人只要拥有驾驶证就可驱车前进，但实际上只有那些有技术和经验的人才能发挥赛车的良好性能。

功能强大和灵活是程序设计语言的主要优势，可以编写出运行在指定机上的需要最少机器资源同时有独特的界面控制和功能的应用程序。

执行速度快是编程语言的另一优势，一个用程序设计语言编写的应用程序是经过编译和连接后生成的机器代码，其执行速度快于靠著作工具解释的应用程序。例如，一个经过编译的程序显示一屏要等待 1/4s，而一个用著作工具作的软件完成同样任务可能要等待几秒，这一差别对最终用户极其明显。

另外，一个应用程序在使用过程中暴露出的问题几乎都可追溯到这个特定的应用程序是如何编写的，而且可由程序员自己解决。同时，既然著作工具本身就是用程序语言编写的，那么它们本身就可能存在缺陷，这种风险也要由用户承担。

采用程序设计语言制作多媒体软件的主要劣势在于人工成本高和开发时间长。一般来



说,一个有经验的程序员要比会用某一著作工具的人员报酬高。一个小型的应用软件,也许让一个并无多少技能的人员采用著作工具即可方便地设计出来。另外,用编程语言设计一个应用程序可能比用著作工具来设计要花费更长的时间,因为程序员必须从头开始,而且要设计一些小的模块。

## 7.2 Authorware 的使用

### 7.2.1 Authorware 简介

#### 1. Authorware 的产生与发展

Authorware 是美国 Macromedia 公司于 1991 年 10 月推出的多媒体制作工具,它可以在 Windows 3.x、Windows 9x、Windows Me、Windows 2000 和 Windows NT 等环境下运行的多媒体制作软件。Authorware 用途广泛、功能强大。因此,得到了广泛的推广和使用。

经过不断的完善和提高,并针对网络技术的发展和应用,Macromedia 公司在 1996 年推出 Attain 版,其后,又先后推出了 Authorware 5.1 和 5.2 版。2001 年 8 月推出了最新版的 Authorware 6.0。

#### 2. Authorware 6.0 的运行要求和集成开发环境

Authorware 6.0 在以前版本的基础上做了较大的改进,因此所要求的软件支持也有所提高。一般为 Windows 9x、Windows Me、Windows 2000 或 Windows NT。

多媒体创作涉及图像、音乐、动画、视频等多种媒体文件,因此要进行 Authorware 6.0 的创作,对计算机的硬件要求较高,其基本配置如下:

- (1) PII 400 的 CPU。
- (2) 64MB 内存。
- (3) 15in 的 SVGA 显示器。
- (4) 至少 2GB 的硬盘空间。

#### 3. Authorware 的主要功能和特点

Authorware 是一个基于流程图的交互式多媒体制作软件。它在流程上设计图标组织程序流程,采用窗口界面和按钮显示方式。使程序的逻辑性强,便于管理。

Authorware 主要具有以下一些功能和特点:

(1) 积木式图标设计,制作交互式应用程序。Authorware 中的每个图标都可以看作是一个模块,在制作多媒体软件时,可以运用这些图标引用一些特定的多媒体元素或进行过程控制。最后,在对这些图标进行组合。这样的制作提高了程序的逻辑性。

(2) 可以直接对演示对象进行编辑处理并具有较强大的文字处理功能。

(3) 具有一些比较基本的图像处理功能,能够实现被显示对象的简单动画。

(4) 具备强大的交互功能。Authorware 具有 11 种交互功能,交互功能十分强大。Authorware 制作的多媒体程序几乎能涵盖了所有的交互类型。

(5) 提供了库和文件的调用。为了精简程序,Authorware 能建立库文件,当不同的程序用到相同的素材时,可到库(Library)中提取。另外,Authorware 允许将整个应用程序分成几个逻辑结构模块,每个结构模块就是一个独立的交互式应用程序,最后由总程序调用,

将这些模块合并便可成为一个多媒体作品。

(6) 自身具有丰富的变量和函数。Authorware 自身具有 300 多个变量和函数。开发者可以通过对这些函数的灵活运用来实现对多媒体程序的控制。

(7) 能实现不同类型程序的动态链接 (DLL)。在编程时, 有一些功能或效果仅用 Authorware 自带的变量和函数实现起来比较困难, 这时可以先用其它语言, 如 VB, C 等事先编好, 然后利用动态链接的功能, 将程序使用动态链接库插入到 Authorware 程序中。

(8) 跨平台运行。不像其它多媒体制作软件, Authorware 允许用户在一个环境下创建, 然后移到另一个平台上继续开发, 实现几乎完全透明的移植。

(9) 一键式出版。执行 File→Publish→Publish 便可将程序打包发布和网络发布合在一起: 保存交互式应用程序后, 发布到 Web、CD-ROM 或局域网。

(10) Authorware 的网络功能。Authorware 中的所有功能, 如动画、交互和数据采集等, 都可以用到 Internet 上。在 Authorware 中采用了 Shockware 技术, 能使经 Shockware 压缩的文件在 Internet 浏览器中进行播放。在最新版 Authorware 中的网络播放器比以前的小 40%。

#### 4. Authorware 的窗口组成。

Authorware 应用程序窗口由标题栏、菜单栏、工具栏、图标栏、设计窗口和流程线等组成。如图 7-1 所示。

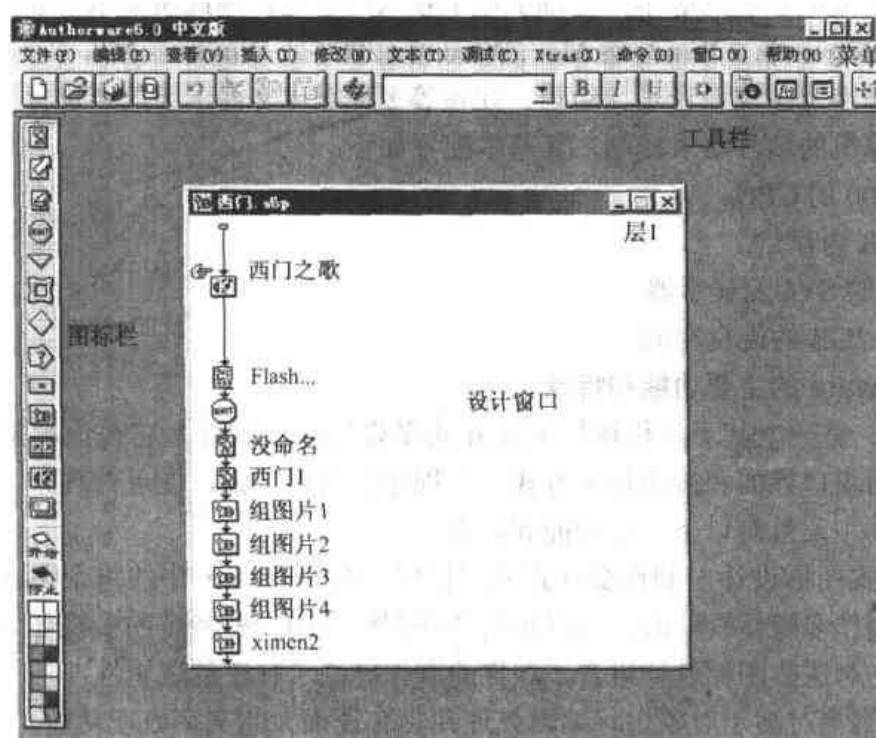


图 7-1 Authorware 应用程序窗口

(1) 标题栏。标题栏是 Authorware 应用程序窗口最上面的一个矩形条, 显示应用程序 (Authorware) 及其商标图案。

(2) 菜单栏。菜单栏位于标题栏的下方, 其中每个选项都代表一个菜单, 而且每个菜单又包括一个命令列表。在 Authorware 6.0 中包括: “File”、“Edit”、“View”、“Insert”、

“Modify”、“Text”、“Control”、“Xtras”、“Commands”、“Windows”、“Help”共 11 个菜单。

1) File 菜单。File 菜单允许用户新建、打开、关闭、保存文件；进行文件、导航、视频等有关设置；打包、出版发行文件；另外还有退出和打印的功能。

2) Edit 菜单。Edit 菜单允许用户撤消或恢复最近一次操作；对象成组和拆组，插入和连接对象；搜索和获取图标信息。

3) View 菜单。View 菜单允许用户查看当前图标、改变窗口的设置以及设置界面功能。

4) Insert 菜单。Insert 菜单可用来引入模板、图标、文字和其它 OLE 对象，还可以改变控制方式。

5) Modify 菜单。Modify 菜单主要用来修改图标、图像、文件等的属性，并对图标及其内容、位置进行编辑修改，还可以改变前景和背景的设置。

6) Text 菜单。Text 菜单允许用户设置文本的字体、大小、字型、对齐方式以及制作滚动窗口等。

7) Control 菜单。Control 菜单主要用来调试程序，具有单步运行、分段运行等功能。

8) Xtras 菜单。Xtras 菜单中提供了库链接设置、拼写检查和音频格式转换等命令。

9) Commands 菜单。Commands 菜单提供了 Authorware 6.0 的在线资源和一些编辑器，用户还可以将自己的指令加入到该菜单中。

10) Window 菜单。Window 菜单用来对演示窗口、库窗口、按钮窗口、控制面板窗口、变量窗口、函数窗口、等进行开关操作。

11) Help 菜单。Help 菜单利用 HTML 的格式，提供了内容详实、功能强大的在线帮助，用户可以十分方便的浏览各部分内容。


(3) 图标栏。图标栏是 Authorware 核心的部分，它由 13 个图标、2 个流程起止标记和调色板组成。具体的使用方法在后面的图标的使用中会详细介绍。


(4) 设计窗口与流程线。设计窗口是编写程序的主要区域。在该窗口内画一条类似流程图的流程线，它用于编辑程序的流程。可以在流程线上加入开发者所需图标，然后在图标上双击，则可编辑各种图标的内容，这样就可以制作出多媒体的应用程序了。


## 7.2.2 图标的功能、使用和素材的灵活引用


### 1. 图标的功能

图标是 Authorware 进行多媒体制作的基本手段，也是最重要的手段。Authorware 中提供了多种功能的图标。其功能如下：

 显示图标 (Display Icon)。用于显示文本、图形、静态图像等。


 移动图标 (Move Icon)。使选定图标中的文字图片动画等产生简单的多种形式的动画。只有具备显示功能的图标才可以被移动。


 擦除图标 (Erase Icon)。用来擦除选定图标中的文字、图片、声音、和视频等，能够产生一定的动画效果。


 等待图标 (Wait Icon)。使程序暂时停下来，直到程序继续运行的条件成立后，程序会自动继续向下执行。


 导航图标 (Navigation Icon)。实现程序流程内的跳转，其作用相当于高级语言


中的 goto 语句。


 框架图标 (Framework Icon)。一般用来建立翻页结构和超文本文件, 其提供了一个复合的功能模块。


 分支图标 (Decision Icon)。用来设置一个逻辑判断结构, 类似于高级语言中的 If……Then……结构。下挂在分支结构下的图标成为分支项, 当程序运行到分支结构时, Authorware 会按照分支的属性设置来决定流程到底沿哪个分支项执行。


 交互图标 (Interaction Icon)。Authorware 中最重要的图标, 用来设置交互结构, 提供用户响应, 从而实现人机交互。Authorware 6.0 中提供了 11 种响应类型, 实现了交互的多样化。


 运算图标 (Calculation Icon)。用于导入函数、设计变量、执行各种运算。


 群组图标 (Map Icon)。利用它可以将一组图标组成一个简单的组。

 数字电影图标 (Digital Movie Icon)。用来引入外部的动画文件, 利用它就可以播放 AVI、FLI、FLC、MOV 等格式的数字电影和动画。

 声音图标 (Sound Icon)。用来引入如: WAV、MP3、VOX、SWA 等格式的声音文件, 并可以对播放方式进行控制。


 视频图标 (Video Icon)。用于在多媒体应用程序中, 控制外接视频播放设备的运行。该类图标的硬件要求较高, 故使用较少。

 开始/结束图标 (Start/Stop Flag)。用于程序的调试, 当使用 start flag 标志旗 Authorware 将从该标志旗处向下执行。当使用 stop flag 标志旗, 正在运行的 Authorware 程序将从该标志旗处停止执行。

 图标调色板 (Icon Color)。用来为图标着色, 可以让程序开发者方便的区分各类图标, 并对程序的执行毫无影响。

## 2. 多媒体素材的灵活引用

Authorware 作为一个非常强大的多媒体制作软件, 它既具有一些文字处理、图像处理 and 实现动画的能力, 又能够灵活自如的引用绝大部分的外部素材, 如图片、文本、声音、视频等, 从而实现了 Authorware 的集众家之所长为我所用的特点。下面介绍如何在 Authorware 中引用素材。

(1) 图片的导入。图片的导入主要是利用  显示图标 (Display Icon), 它主要用于显示文本、图形、静态图像。

在引入图片时只需将一个显示图标拖到流程线上, 然后给该图标起个名字, 双击该图标, 进入编辑窗口, 执行菜单命令 File→Import, 这时会弹出 “Import Which file?” 对话框, 在对话框中选择要导入的图片即可。

这样导入的图片在出现时会显得有些单调, 我们可以利用转场效果为图片的出场增加效果。双击图片, 将 Authorware 切换到编辑状态, 图片周围出现 8 个控制点, 表示进入编辑状态。选择菜单 Modify→icon→Transition (或者直接按 (Ctrl) + (T) 组合键), 系统弹出 Transition 转场对话框, 如图 7-2 所示。

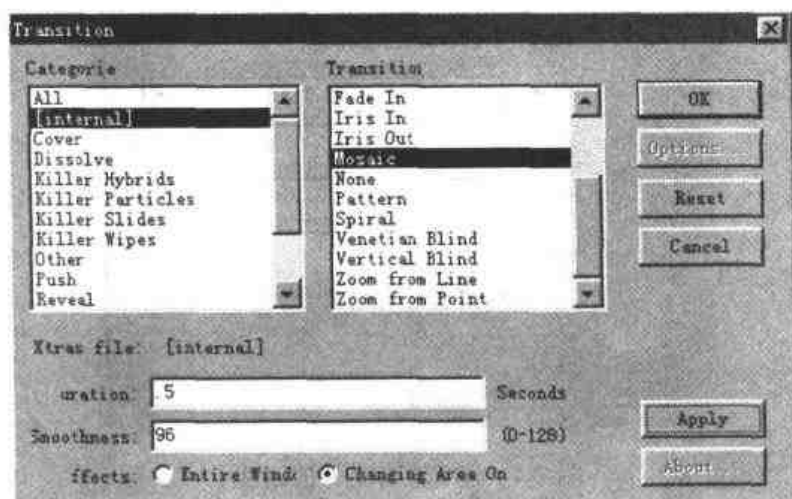




图 7-2 Transition 转场对话框

若在 **Categorie**（转场类别）中选择 **internal** 内部转场，可以在右边该类包含的 **Transition** 效果中选择 **Mosaic**（马赛克）效果。单击【**Apply**】按钮就可以看到图片的出场效果是马赛克出现的。按【**OK**】确认，再单击运行 ，可以看到图片出场的效果有了变化。

（2）音频文件的引入。音频的导入主要是利用  声音图标（**Sound Icon**）来进行的。可以引入的文件类型有 **WAV**、**MP3**、**VOX**、**SWA** 等格式的声音文件（其中引入 **MP3** 是 **Authorware6.0** 的一项新增功能），并可以对播放方式进行控制。

同样，在导入已编辑好的声音文件之前，首先要在流程线上插入一个声音图标。双击声音图标，系统弹出如图 7-3 所示的声音文件选择框。

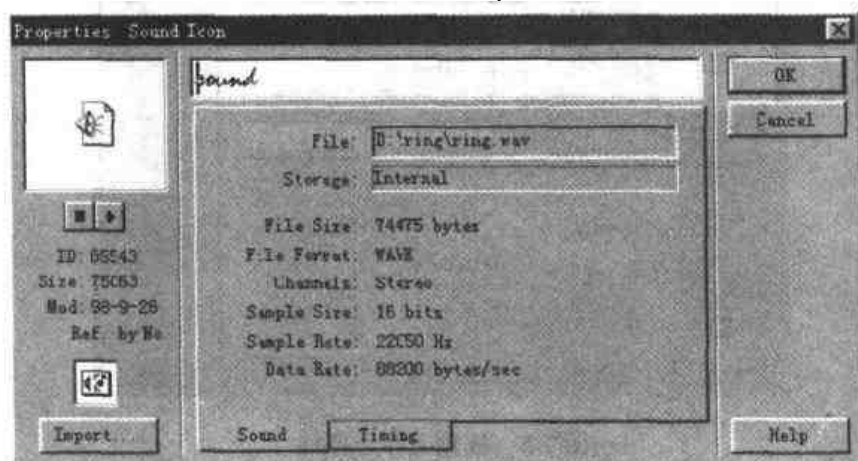


图 7-3 声音文件选择框

单击【**import**】按钮，导入一个声音文件。运行程序即可听到导入的声音文件。图 7-4 是对声音文件属性的设置。

单击 **Timing** 标签，可以对声音的播放进行设置，其中：

**Concurrency** 中有三个选项：**Wait Until Done**（等到声音播放完再往下执行程序），**Concurrent**（声音执行的同时继续往下执行程序），**Perpetual**（永久，当 **Authorware** 退出声音图标以后声音图标依然激活。也就是说可以设置一个条件，以后条件一满足就播放声音）。

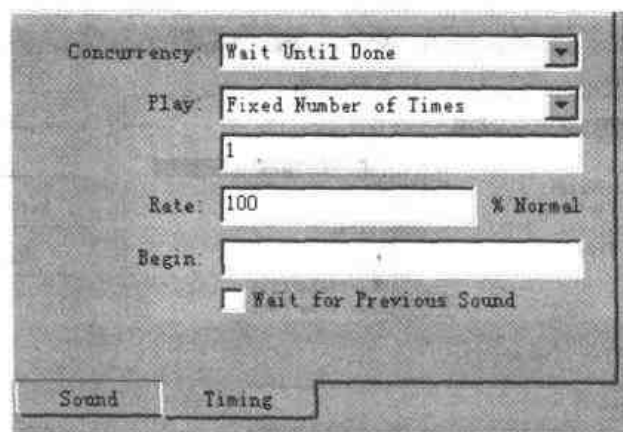



图 7-4 设置声音文件的属性

**Play** 设置声音的播放次数。Fix Number of times 指定声音播放次数。下面框中的 1 表示播放一次，可以自行设置播放次数。Until True 表示直到条件为真停止播放音乐。

**Rate:** 设置声音播放速度。

**Begin:** 设置声音播放条件何时开始播放。

**Wait for previous Sound:** 等待前一个声音播放完毕。

(3) 视频文件的导入。在流程线上插入一个  数字电影图标 (Digital Movie Icon)，它是用来引入外部的动画文件，利用它就可以播放 AVI、FLI、FLC、MOV 等格式的数字电影和动画。

双击电影图标，系统弹出文件选择框如图 7-5 所示。

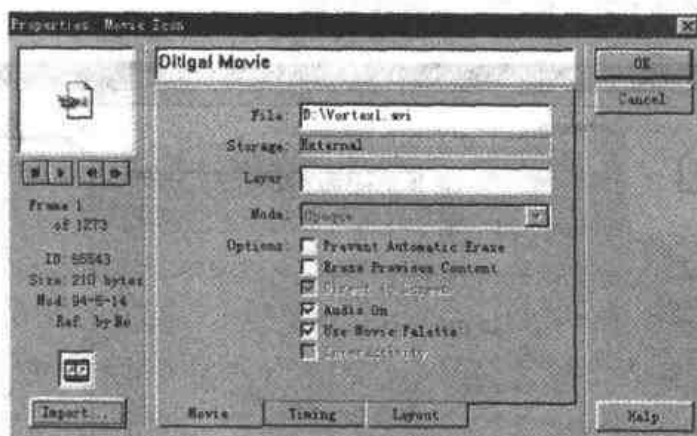


图 7-5 视频文件选择框

单击 **Import** 按钮，即可导入一个电影文件。运行程序可以看到导入的电影文件。关于数字电影的 Timing 具体参数同前面的声音设置。

(4) flash 的播放。Flash 是当今最为流行的二维动画制作软件，在 Authorware 中经常会遇到播放 flash 的情况，下面简单的介绍一下将 flash 引入 Authorware 的方法。

在菜单中执行 **Insert→Media...→Flash**，会弹出 flash 设置窗口，如图 7-6 所示，单击 **浏览** 按钮选择要引入的 flash 文件即可。

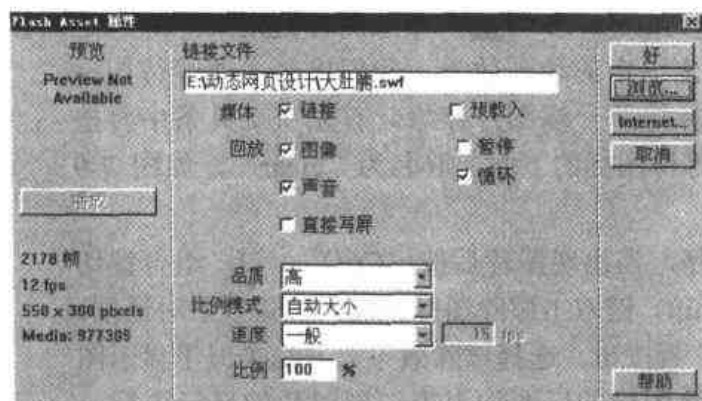



图 7-6 flash 设置窗口

以上四种导入多媒体素材的方法是 Authorware 中最常用也是最重要的。

## 2. Authorware 中其他图标的功能和使用

(1) 等待图标 (Wait Icon) 。等待图标可以使程序暂时停下来，直到程序继续运行的条件成立后，程序会自动继续向下执行。

在制作图片欣赏软件时，程序是由多幅图片组成。屏幕应该出现下一张图片，同时前一张图片消失。要图片消失就要加上擦除图标，在擦除图标的后面跟上一张图片的显示图标。但是事情并不是这样简单，因为如果按照这样排放流程线的话，运行结果一定是图片一闪而过，马上出现第二幅图片。因为显示图标的后面马上是擦除图标缺乏等待的时间。因此我们需要用到等待图标。

首先按照前面介绍的导入图片的方法导入一幅图片，起名为“pic1”。在工具条中将等待图标拖曳到流程线中显示图标的下方，起名为“wait”。如图 7-7 所示。

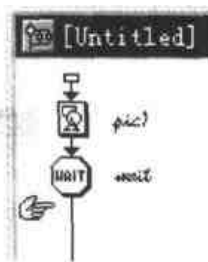


图 7-7 使用等待图标

双击流程线上的等待图标，系统弹出如图 7-8 所示的等待图标对话框。

Events 为事件选择，Mouse Click 是鼠标单击，Key Press 为按任意键，Time Limit 时间限制设置为 5s，选中 Show Button。在运行程序时，第一幅图片运行后屏幕上将出现 Continue 按钮。按鼠标、按任意键、按按钮都可以继续程序，或者 5s 以后程序自动继续执行。

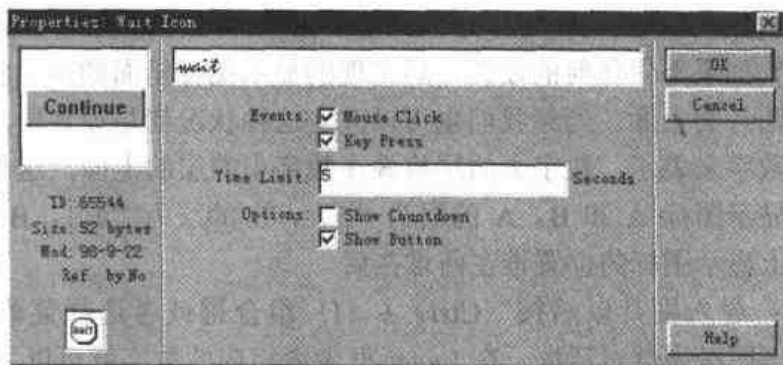



图 7-8 等待图标对话框

(2) 擦除图标 (Erase Icon) 。擦除图标是用来擦除选定图标中的文字、图片、声音和视频等，能够产生一定的动画效果。我们接着上面等待图标的例子，使用擦除图标将第一幅图片擦除。首先从工具条中将擦除图标移动到流程线等待图标的下面，取名为“clear”，如图 7-9 所示。

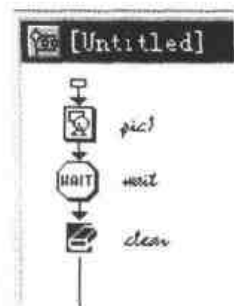




图 7-9 使用擦除图标

下面运行一遍程序，当出现图片后按 (Ctrl) + (J) 组合键切换到流程线。按住 (shift) 键双击擦除图标，进入擦除效果对话框，单击 Transition 右边的  按钮。选择擦除效果。方法类似于显示图标的显示效果。在擦除图标的后面加上第二副图片的显示图标。运行程序，可以看见第一幅图片先显示然后按键以后第一幅图片擦除，显示第二幅图片。这样便可以实现一个简单的图片欣赏的程序了。

(3) 移动图标 (Move Icon) 。移动图标主要用于选定图标中的文字、图像对象等产生简单的多种形式的运动效果。只有具备显示功能的图标才可以被移动。

首先在流程线上插入一个显示图标，并导入一幅图片作为移动的对象。运行一下程序，按住 (shift) 键，双击移动图标，系统弹出移动图标对话框，如图 7-10 所示。在移动的类型里面选择“Path to End”路径移动，单击选中要移动的图片，将其移动到屏幕的另一端，运行的时间 Timing 默认为 1 秒。如果觉得太短可以自行设置，设置完成后单击【OK】按钮确定。

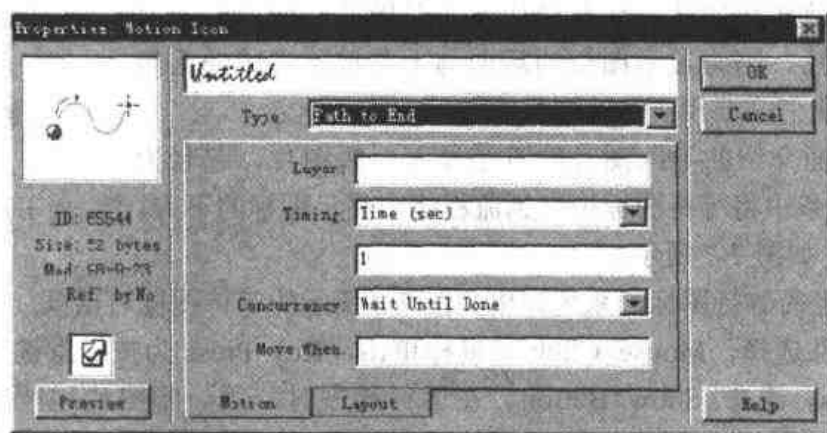



图 7-10 移动图标对话框

层的设置：当我们在 Authorware 中导入图片的时候，往往会碰到两幅图片重叠的情况，谁在上，谁在下呢？如果不做任何的设置，后出现的显示图标里面的图画将会重叠在先出来的显示图标里面的图片的上面。当然我们也可以改变这种状况就是设置层。层是显示图标的一种属性，用一个数字来表示。数字大的层将叠在数字小的层的上面，这种比较是相对比较。比方说有两个显示图标 A 和 B。A 的层为“5”，B 的层为“4”，B 就显示在 A 的下面。不管在流程线上显示图标的位置谁在前谁在后。

层的设置方法：导入图片以后按 (Ctrl) + (I) 组合键或者选择菜单 Modify → icon → properties，系统弹出图标属性对话框。在 Layer 框中填相应的数字就可以了。


(4) 群组图标 (Map Icon) 。利用它可以将一组图标组成一个简单的组，组图标虽然使用起来很简单，但它可使流程看上去更加清楚。




分组时有两种方法：

1) 把图标打包成组选中要分组的图标。按〈Ctrl〉+〈G〉组合键就可以打包成组，双击组可以看到包在里面的图标。

2) 在工具栏中将组图标拖动到流程线上，双击组图标，打开窗口。将图标拖入窗口内流程线。


(5) 框架图标 (Framework Icon) 。用来建立翻页结构和超文本文件，提供了一个复合的功能模块。

一般用框架图标进行组织的整个流程。所有的功能分成一个个小的组，把组放在框架图标也就是像镜框一样的那个图标的右边。在这里一个组就像是书的一页，可以前后翻。运行整个程序的时候画面中会出现导航面板。

(6) 交互图标 (Interaction Icon) 。这是 Authorware 中最重要的图标，用来设置交互结构，提供用户响应，从而实现人机交互。Authorware 6.0 中提供了 11 种响应类型，实现了交互的多样化。

11 种响应类型有按钮交互、热区交互、热件交互、目标区域交互、菜单交互、条件交互、文本输入交互、按键交互、尝试次数交互、时间限制交互、事件交互。

下面举一个简单的交互流程的示例。从图 7-11 中可以看到，当运行到交互这一个图标的时候，系统就等待用户进行某个（在这里是单击按钮）操作。当用户按到了【子功能】这个按钮的时候，程序就执行下面黑色的群组图标里面的程序。整个过程就是用户发出操作→执行相应操作对应的程序。本图片展示的是一个按钮交互。就是按下【子功能】按钮，执行下面的程序。

(7) 分支图标 (Decision Icon) 。分支图标用来设置一个逻辑判断结构，下挂在分支结构下的图标称为分支项，当程序运行到分支结构时，Authorware 会按照分支的属性设置来决定流程到底沿哪个分支项执行。

分支图标有很多种形式，图 7-12 所示的是简单的循环形式。分支图标下面挂了三个群组图标，运行的时候会从左到右依次执行。

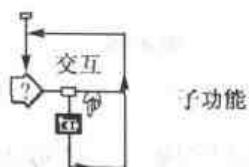


图 7-11 交互流程的示例

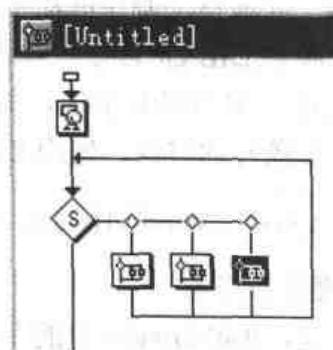


图 7-12 简单的分支

双击分支图标可以对分支图标进行设置。将 Repeat（重复）设置成为 Until All paths Used（直到所有的分支都用过），Branch（分支）设置成为 Sequentially（顺序执行每一个分支）。

### 7.2.3 Authorware 的高级制作

在 Authorware 的整个开发过程中,通过图标可以完成简单的多媒体制作,但一般来说,要想完成一个好的多媒体作品,单单靠图标级的实现是不够的。因此,需要使用语句方式去控制所要处理的数据,这种情况经常出现在需要动态装入某种数据格式的文件,而不是事先预装好文件。这时,可以使用标准的外部函数动态调换数据来实现,如 MIDI 文件播放、动态交换调色板、动态装入外部点位图及动态数据交换(DDE)等,都属于这一范畴,也就是 Authorware 的高级制作。此外,当制作一些具有特殊功能的应用程序时,还会有 5%左右的功能即使利用标准外部函数也无法完成。这时,便需要通过 Authorware 与高级语言的接口完成自定义外部函数的制作。因此,凡是在这一层次上完成的功能都被称为自定义接口级实现。在开发 Authorware 应用程序时,要明确自己在上面三个层次中的哪个位置。若能够以图标级实现,就不要以标准外部函数级实现,若能够以标准外部函数级实现,就不要以自定义函数实现。

#### 1. 动态交换调色板

在 256 色显示方式下,系统同时只能有一个激活的调色板,这样对于许多调色板不统一素材的处理方案可分为两类:一类是不同的素材统一调色板以适应单一的显示系统,这种方法是素材制作工具方面的应用;另一类是在程序运行过程中切换外部函数实现更换调色板的功能。

这两种处理方法在具体开发过程中经常是同时使用。特别是在素材文件较多,而且颜色基调相差很大的情况下,很难统一调色板或统一后效果不理想,这时就需要使用第二种方法。Authorware 在处理调色板时是以 Microsoft RIFF 格式的调色板文件为标准,无论是哪一种格式的素材都需要提取调色板信息,并存成 PAL 文件格式。利用 Authorware 外部函数 palette.ucd 中的 SetPalette() 函数对调色板进行设置。

#### 2. 动态装入外部点位图

这个功能在需要更换大量的尺寸类似、但内容不同的图片素材的情况下经常使用,以节省主程序的空间,并使整个制作过程不过于机械化。在需要引入外部图片的节点处,通过计算图标调用外部函数 DIB.UCD。这个库中有两个函数:一个是 DisplayDIB;一个是 EraseDIB。这两个函数在使用时需要成对出现,即当显示一个外部图片时,需要用 DisplayDIB;而当需要删除这个外部图片的时候,使用 EraseDIB 函数。其中 DisplayDIB 的调用格式是:

```
id:=DisplayDIB(WindowHandle, "MyFile.Dib", Left, Top, UsePalette)
```

#### 3. 其他标准扩充函数

在前面曾讲过, Authorware 中的有些功能既可以在函数级实现,如播放 WAV 文件、播放 3DS 动画;也有些功能必须通过函数实现,如播放 MID 文件、播放音乐 CD 等。这些功能都通过标准库函数 apwmme.ucd 完成。其实,这个库中的函数可以控制任何一个 MCI 设备。当有大量的声音文件需要播放时,采取函数级播放可能会获得较高的制作效率和较好的扩充性能,因此函数级作为对图标级制作的补充是必不可少的。除 apwmme.ucd 以外,还有一个 dde.ucd。这个函数库的功能也很强, Authorware 通过这个函数库可以和其他 windows 应用程序完成动态数据交换。这个函数库的一个典型应用是制作一个光盘应用的安装程序。

这通常是整个光盘的最后一步，利用 Authorware 与文件管理器的 DDE 交换，完成组建与建项功能。此外，Authorware 还带有一个内存存取函数库 awitools.ucd，这个函数可以完成对内存地址的操作。

#### 4. Authorware 功能扩展接口的实现

Authorware 的功能扩展是以 Windows 动态链接库方式实现的。通过编程接口可以实现两类扩展功能：图像处理功能和数据处理功能。下面将分别对这两类进行介绍：

(1) 图像处理。在 Authorware 制作的应用中添加特殊的图像处理（如动态装入外部图片）需要使用此种接口方式。因为 Authorware 对 Presentation Window 中的各个物体都是统一管理的，所以若要在 Authorware 管理的 Presentation Window 中添加自定义的物体，必须理解 Authorware 的屏幕接口管理机制。

在 Authorware 的屏幕下登录一个对象的过程是由 APW 屏幕管理器来管理的。对象过程本身是由对象自身提供的，并存于 DLL 文件中。对象过程接收发自 APW 屏幕管理器的消息，并进行相应的处理操作。一旦自定义物体登录在 APW 屏幕管理器中，此物体便可与 Authorware 中的其它正常物体一样做各种操作，如对它动画或用删除图标删除此物体。

在对象过程与 APW 屏幕管理器的消息传递过程中，存在两类消息方式：通知消息和控制消息。通知消息是由 APW 屏幕管理器发出的，对象过程通过此类消息后做出相应处理。一个对象在 APW 中的存在周期从登录开始到登录结束或被删除或程序结束。消息传递是按下面的方式实现的：

对象过程首先发出 APWC-POST 消息给 APW 屏幕管理器，完成自定义物体的登录过程（有关 APWC-POST 消息及其相关结构的定义均在 APWPOST.H 中）。

在建立完 APWC-POSTPB 结构所需要的数据后，使可向 APW 管理器发送 APWC-POST 消息。

当在 APW 中登录该物体成功后，Authorware 将返回分配给该物体的 ID 值。另一个重要的过程函数便是对象函数，此函数将完成接收 APW 发来的通知消息，并需要对各种消息进行相应的处理。对象函数必须是 APW-PROC 函数型。对象函数将接受 APWN-AB-PTR 和消息标号两个参数。

其中，APWN-AB-PTR 为指向参数块的指针。消息标号是通知消息标识符。在对象函数中，将处理各种消息，如 APWN-INIT, APWN-DESTROY, APWN-PAINT, APWN-SAVE, APWN-RESTORE 等消息。

APWN-PAINT 是其中最重要的消息，它将真正完成所登录的物体在屏幕上以什么内容显示出来（有些类似于 WM-PAINT 消息）。

APWN-DESTROY 将完成释放私有数据所占用的内存。

(2) 数据处理。由于数据处理不涉及 Authorware 的内部管理机制，因此接口相对比图像处理要容易实现。只要保证函数调用时参数的数据类型一致和返回值类型一致即可。

如果已有一个独立的应用程序完成某一个数据运算，那么将其改为接口程序是非常简单的。只要增加一个 LibMain 动态库主入口函数和 WEP 出口函数，并建立相对应的 RC 文件以指明在装入函数时参数的类型，就可以很方便的在 Authorware 中调入该函数。

## 7.2.4 Authorware 文件的打包和发行

在 Authorware 编辑完成的程序系统在未打包和发行之前, 这些程序只能在 Authorware 环境下运行。一般 Authorware 的开发商交给用户能够在 Windows 环境下运行而且无法对其进行编辑的产品。Macromedia 公司为了适应这类媒体发行商的需求, 使得 Authorware 具有文件打包的功能, 经过打包后的文件是一个可执行文件, 不含应用程序的源代码。

### 1. 打包发行时包含的文件

一个典型的 Authorware 应用程序一般不但包含应用程序本身, 而且还要有一些外部文件。因此要让用户能正常运行这个软件产品, 通常应包括以下文件:

- (1) RunA16 (32) 执行控件。
- (2) 应用程序中有链接的外部文件, 如声音文件、数字化电影文件等应用程序文件, 这些文件可在 Ayuthorware 6.0 的目录下找到。
- (3) 用于 Quick Time for Windows、Video for Windows, 视频重叠卡和录像片的驱动程序。
- (4) 若在编辑程序时用了一些特殊的字体, 这种字体在用户的计算机上可能没有安装, 所以, 还应有特殊字体的字库。
- (5) 编辑程序时所用到的动态链接库 DLL 和外部函数文件。
- (6) Xtras 文件, 如 Active X 控件、Transition Xtrax 效果文件。

由于 Xtras 文件和外部函数文件都不是很大, 因此可以将它们全部都复制过来。这样还有一个好处, 便于日后添加控件或修改函数。

### 2. 特殊说明

除了具有相应的驱动程序外, 还要附加一些额外的支持:

- (1) Quick Time 数字化电影: 需要 a6qt.xmo 或 a6qt32.xmo 支持。
- (2) AVI 数字化电影: 需要 a6vfw.xmo 或 a6vfw32.xmo 支持。
- (3) Director 数字化电影: 需要 a6dir.xmo 或 a6dir32.xmo 和 director 子目录。
- (4) RunA6W16 (32) 执行控件、部分 UCD、Xtras 文件和数字化电影额外支持文件在 Authorware 6.0 的安装目录下均有。
- (5) 特殊字符的字库文件在路径 “c:\\WINDOW\\FONTS” 下。
- (6) 其他的一些驱动程序和外部函数可到专业的 Authorware 网站下载。
- (7) 在打包时只要把这些外部文件与要创建的可执行文件在同一目录下即可。

下面是对制作的程序进行打包的操作过程:

打开做好的程序。选择 File→Package... 系统弹出对话框。

Package File 有三个选项:

- “Without Runtime”。用这种方式打包的文件不包括执行控件, 即不形成 EXE 文件, 而是 A4R 文件, 它需要由 RunAW16.exe 或者 RunAW32.exe 来调用执行。
- “For Windows 3.1”。16 位的操作系统 (如 Windows 3.1) 只能回放用这种方式打包形成的 EXE 文件 (Windows 9x 或者 NT 等 32 位操作系统也能接收这种文件, 但性能减低)。
- “For Windows 9x and NTvariant”。用这种方式打包形成的 EXE 文件可以在 Windows

9x 或者 NT 这样的 32 位操作系统环境下运行。

“Resolve Broken Links at Runtime” 选项：选中后，当用户运行程序时将恢复断开的链接。

“Package All Libraries Internally” 选项：选中后，将使所有与当前交互式应用程序有链接关系的库文件成为打包文件的一部分，否则将单独打包：一般不选中该项，因为库文件只有在需要用的时候才被调用，所以将文件单独打包可以提高应用程序效率。

“Use Default Names When Package” 选项：选中后，将使打包后的文件与当前的应用程序文件同名。

选择 For Windows95, 98 and NT。

如图 7-13 所示。单击【Save Files & Package】，给定 exe 文件名，程序完成。打包以后不能独立运行，还要在 exe 的目录中放上 Xtras 目录。因为 Authorware 制作的程序需要这些文件来支持。Xtras 目录在 Authorware 目录中。



图 7-13 打包

一个完整的打包文件应包括的文件如下：

-你的源文件.a6w	;建议在制作之前就规划好!
-你的打包文件.exe	
- *.*	;你的 Authorware 安装目录下
-Xtrax - *.*	;的 Xtrax 目录及以下文件。
- *.*	;具体文件可用命令菜单下的
	;查找 Xtrax 命令。把整个目录
	;复制比较省事!
-*.xmo	;播放数字化电影等的额外支持文件
-*.U32	;你的 Authorware 安装目录下
-*.DLL	;的相关文件
-*.UCD	;所有的媒体不同，则文件会
	;有所不同!
-IMAGES (目录)	;用于存放你的图片
-MEDIA (目录)	;用于存入你的动画、视频日后添加
	;控件或修改函数。

## 7.3 小结

1. 多媒体著作工具一般都提供了不同的应用开发环境，并具有各自的功能特点，使用于不同的应用范围。根据多媒体著作工具的创作方法和特点的不同，可将其划分为如下四类：

- 基于时间的著作工具。
- 基于图标或流程线的著作工具。

- 基于卡片或页面的著作工具。
- 以编程语言为基础的著作工具。

2. Authorware 是一个基于流程图标交互的多媒体制作软件。它在流程上设计图标组织程序流程，采用窗口界面和按钮显示方式，使程序的逻辑性强，便于管理。

## 第8章 多媒体数据库及基于内容检索

### 本章要点

- 多媒体数据库的体系结构（组成、存储和管理）。
- 基于内容检索系统的体系结构和关键技术以。
- 基于内容检索系统的设计原理和实现技术。

随着计算机技术的发展，计算机越来越多地用于信息处理，如财务管理、办公自动化、工业流程控制等。这些系统所使用的数据量大、内容复杂，而且面临数据共享、数据保密等方面的需求，于是便产生了数据库系统。数据库系统的一个重要概念是数据独立性。用户对数据的任何操纵（如查询、修改）不再是通过应用程序直接进行，而是必须通过向数据库管理系统（DBMS）发请求实现。DBMS 统一实施对数据的管理，包括存储、查询、处理和故障恢复等，同时也保证能在不同用户之间进行数据共享。如果是分布数据库，这些内容将扩大到网络范围之上。

依据独立性原则，DBMS 一般按层次被划分为三种模式：物理模式、概念模式和外部模式（也叫视图）。物理模式的主要职能是定义数据的存储组织方法，如数据库文件的格式、索引文件组织方法、数据库在网络上的分布方法等。概念模式定义抽象现实世界的方法。外部模式又称子模式，是概念模式对用户有用的那一部分。概念模式通过数据模型来描述，数据库系统的性能与数据模型直接相关。数据模型的不断完善和变革，也就是数据库系统发展的历史。数据库数据模型先后经历了网状模型、层次模型、关系模型等阶段。其中，关系模型因为有比较完整的理论基础，“表格”一类的概念也易于被用户理解，因而逐渐取代网状、层次模型，在数据库中居主导地位。关系模型把现实世界事物的特性抽象成数字或字符串表示的属性，每一种属性都有固定的取值范围。于是，每一个事物都有一个属性集及对应它的属性值集合。

近年来，随着多媒体数据的引入，对数据的管理方法又开始酝酿新的变革。我们知道，传统数据库模型主要针对的是整数、实数、定长字符等规范数据。数据库的设计者必须把真实世界抽象为规范数据，这要求设计者具有一定的技巧，而且在有些情况下，这项工作会特别的困难。即使抽象完成了，抽象得到的结果往往会损失部分原始信息，甚至会出现错误。当图像、声音、动态视频等多媒体信息引入计算机之后，可以表达的信息范围大大扩展，但又带来许多新的问题。因为多媒体数据不规则，没有一致的取值范围，没有相同的数据量级，也没有相似的属性集。在这种情况下，如何用数据库系统来描述这些数据呢？表格还适用吗？另一方面，传统数据库可以在用户给出查询条件后迅速地检索到正确的信息，但那是针对使用字符数值型数据的。现在，我们面临着这样的问题：如果基本数据不再是字符数值型，而是图像、声音，甚至视频数据，那将怎样进行检索？如何表达多媒体信息的内容？该如何组织这些数据呢？查询该如何进行呢？这些都是值得考虑的问题。

## 8.1 多媒体数据库

### 8.1.1 多媒体数据的存储问题

人们对文本透彻理解、广泛应用已经有很长一段时间了，而多媒体存储则是较新的议题。多媒体存储有一些新的需要考虑的问题：巨大的存储空间、大型对象、多个相关对象、对检索时间的要求等等。我们在此简单看一下其中的一些问题：

#### 1. 巨大的数据量

统计结果表明，我们所使用的全部重要信息中只有不到 20% 的部分被自动化了，而其余 80% 以上的信息一般是写在纸上，或在会议、讨论、演示中进行了交互。纸、胶片或磁带上的记录很难进行集成、控制、搜索、存取和分发。若要确定某一纸张文档、胶片、音频或视频带的位置，需要在只被少数关键人员（这成为信息流中的瓶颈部分）所掌握的众多存储文件、复杂索引系统中搜索，并要进行很好地组织，以确保按正确的顺序放回到最初的存储位置。比定位更复杂的是为文档、胶片和磁带建索引，尤其是当这些不同的媒体结合成一个单一的多媒体文档时。

#### 2. 存储技术

理想情况是，来自于纸张、胶片、音频和视频带以及直接来自摄像机的信息可以用处理数据、文本和图形的同一计算机化信息系统进行管理。其结果是一种集成的战略信息库，它可以被许多人同时快速方便地访问。而实践上要达到这一点，需要将各种存储机制放在公共的存储和检索管理之下。

显微照片和微缩胶片是用作存储纸张文档的一种媒体，但这种方式的确非常麻烦，速度慢并且会频繁出故障。恢复胶片的物理损伤是非常费时的，还会造成信息损失。显微照片和微缩胶片易于发生很严重的机械故障，以及微缩胶片媒体的物理（光化学）变形。要注意的另一点是，显微照片或微缩胶片容易在文档上留下很多噪声（如很小的黑点），因此，当从显微照片或微缩胶片转换到光存储系统（激光盘）上的压缩图像时，这种噪声会造成明显的压缩问题，导致极差的压缩率。致使 A 型（8.5in×11in，信纸尺寸）大小的图像，其平均的 CCITT Group 4 压缩文件的大小可以从 60~70KB，上升到超过 200KB。这种情况下，噪声是非常明显的，并会干扰从显微照片或微缩胶片上印出的文档。显微照片或微缩胶片要求特殊的空调存储条件，并不能为档案文档提供快速或方便的随机存取性能。

有两种主要的大型存储技术可以用于多媒体文档的存储：光盘存储系统和高速磁存储。显然，管理光盘库中的几个光盘要比管理范围更大的磁盘容易许多。维护这样一个系统的运行是很麻烦的。另一个要注意的重要因素是光盘存储对老的、不常用的文档进行脱机归档来说，是一种很好的工具，可保存很长时间。

#### 3. 多媒体对象存储

存储于光媒体中的多媒体对象要实际发挥作用就需要快速自动地定位。这里的一个关键问题是对超媒体文档或超媒体数据库记录进行随机存取。光媒体可提供非常密集的存储。例如，一个 12in 光盘可存储 6.5GB 的信息，这相当于大约 12h 的压缩视频或多达 128000 张以 CCITT Group 4 压缩格式压缩后的 A 型尺寸图像。注意，在全运动视频和音频中，还有第三



维——时间,对压缩数据的容量有主要的影响。一段典型的经压缩的8位声音片断要求50KB/s的数据率,对于16位的声音(频率范围与乐声标准接近),它的数据率将显著地提高到250KB/s,32位声音的要求更高。类似地,一幅视频图像在分辨率低于EGA分辨率(640×350像素)时要求1.5MB/s,对于XGA或HDTV(1280×1024像素,或1125线)分辨率,它将增加到1MB/s,视频剪辑为UDTV水平(3000线)时,要求更大的容量。

检索速度是另一个主要考虑的方面。检索速度与以下几个因素直接相关:存储等待时间(从存储介质中检索数据所需的时间)、相对于显示器分辨率的数据容量(压缩效率)、传输介质和速度(传输等待时间)和解压缩效率。对于快速信息检索来说,索引是必须的。

应用程序通常需要通过许多域访问文档。举个例子,一个犯罪分子的个人记录应通过其社会安全号码或指纹(指纹可能是通过他的社会安全号码定位的)、法庭记事号码、文件标识、名字、别名等来访问,把多媒体文档(如逮捕令、指纹、犯罪时图像、视频片断等)与犯罪嫌疑人相联系可能要求有能力记录这些关系以便把法庭审讯的多媒体文档加入数据库。在属性与图像、音频片断或视频片断的标识之间建立联系的过程称为索引。

#### 4. 多媒体文档检索

标识多媒体文档的最简单形式是通过存储盘标识和文档在盘上的相对位置(文件号)来进行。这些对象可以分组后存在数据库夹中(类似于纸文档存在文件夹中),或存在显示超媒体文档的复合对象中。这是在大多数多媒体系统中标识图像的基本方法。

用标识符访问存在数据库中的对象,这一功能要求数据库能完成所要求的多媒体对象目录功能。对于声音和全运动视频来说,另一个重要应用是能剪切它的某些部分,并将它们与另一段相结合。例如,一段语音或演示可以被剪下来用作另一个RFT文档中的引述。为使声音或视频素材能以参考单元的形式存储,并为许多用户使用,为它建索引的能力就变得很重要,因为这样可以避免在几个小时长的、缓慢运动的带子中搜索这样乏味的工作。当然,新的基于内容的检索技术可以提供更为智能的支持。

### 8.1.2 多媒体数据的管理

#### 1. 传统的数据管理

传统的数据库系统有三种类型:关系型、层次型和网络型。Codd关于关系数据库的开创性工作,建立了关系数据库的坚实理论基础,给出了清晰的规范说明,加上“表格”的概念直观易懂,使得关系数据库在理论和产品开发上都获得了巨大的成功,在数据库市场上占有明显的主导地位,特别是中小型数据库系统。

关系数据库采用关系框架来描述数据之间的关系,通过把数据抽象成不同的属性和相互的关系,建立起数据的管理机制。例如某公司用关系数据库管理雇员的资料。雇员的信息可以抽象为工号、姓名、年龄、性别、月工资、所在部门、该部门的经理等多项属性。按关系模型的要求,雇员信息可以用两个关系表示:雇员(工号、姓名、年龄、性别、月工资、部门编号)、部门(部门编号、部门名称、部门经理)。这两个关系就可以支持关于雇员的检索和查询工作。这个例子说明,对于一个具有复杂结构的实体(如雇员),关系数据库需要把它分解,分解的结果可以用最简单实用的关系(如雇员和部门)表示。实体的结构语义隐性地包含在两个关系的相同属性(部门编号)中。只有通过联结(join)、投影(project)等操作才能体现出结构语义。关系数据库的这一特性非常简洁,既可以用数学理论加以规范

和证明，又通俗易懂，易于被人们接受。

## 2. 多媒体带来的问题

在传统的数据库中引入多媒体的数据和操作，是一个极大的挑战。这不是一个只要把多媒体数据加入到数据库中就可以完成的问题。传统的字符数值型数据虽然可以对很多的信息进行管理，但由于这一类数据的抽象特性，应用范围毕竟十分有限。为了构造出符合应用需要的多媒体数据库，必须要解决从体系结构到用户接口一系列的问题。多媒体对数据库设计的影响主要表现在以下几个方面：

(1) 数据量巨大且媒体之间量的差异也极大，从而影响数据库的组织 and 存储方法。如动态视频压缩后每秒仍达上百 KB 的数据量，而字符数值等数据可能仅有几个字节。只有组织好多媒体数据库中的数据，选择设计好合适的物理结构和逻辑结构，才能保证磁盘的充分利用和应用的快速存取。数据量的巨大还反映在支持信息系统的范围的扩大。应用范围的扩大，显然不能指望在一个站点上就存储上万兆字节的数据，而必须通过网络加以分布，这对数据库在这种环境下进行存取也是一种挑战。

(2) 媒体种类的增多就增加了数据处理的困难。每一种多媒体数据类型都要有自己的一组最基本的概念（操作和功能）、适当的数据结构和存取方法以及高性能的实现。但除此之外也要有一些标准的操作，包括各种多媒体数据通用的操作及多种新类型数据的集成。虽然前面列出了几类主要的媒体类型，但事实上，在具体实现时往往根据系统定义、标准转换等演变成几十种媒体格式。不同媒体类型对应不同数据处理方法，这便要求多媒体 DBMS 能不断扩充新的媒体类型及其相应的操作方法。新增加的媒体类型对用户应该是透明的。

(3) 数据库的多解查询。传统的数据库查询只处理精确的概念和查询。但在多媒体数据库中非精确匹配和相似性查询将占相当大的比重。因为即使是同一个对象若用不同的媒体进行表示，对计算机来说也肯定是不同的；若用同一种媒体表示，如果有误差，在计算机看来也是不同的。与之相类似地还有诸如纹理、颜色和形状等本身就不易于精确描述的概念，如果在对图像、视频进行查询时用到它们，很显然是一种模糊的、非精确的匹配方式。对其他媒体来说也是一样。媒体的复合、分散、时序性质及其形象化的特点，注定要使数据库不再是只通过字符进行查询，而应是通过媒体的语义进行查询。然而，我们却很难了解并且正确处理许多媒体的语义信息。这些基于内容的语义在有些媒体中是易于确定的（如字符、数值等），但对另一些媒体却不易确定，甚至会因为应用的不同和观察者的不同而不同。

(4) 用户接口的支持。多媒体数据库的用户接口肯定不能用一个表格来描述，对于媒体的公共性质和每一种媒体的特殊性质，都要在用户的接口上、在查询的过程中加以体现。例如对媒体内容的描述、对空间的描述以及对时间的描述。多媒体要求开发浏览、查找和表现多媒体数据库内容的新方法，使得用户可以很方便地描述他的查询需求，并得到相应的数据。在很多情况下，面对多媒体的数据，用户有时甚至不知道自己查找的是什么，不知道如何描述自己的查询。所以，多媒体数据库对用户的接口要求不仅仅是接收用户的描述，而是要协助用户描述出他的想法，找到他所要的内容，并在用户接口上表现出来。多媒体数据库的查询结果将不仅仅是传统的表格，而将是丰富的多媒体信息的表现，甚至是由计算机组合出来的结果“故事”。

(5) 多媒体信息的分布对多媒体数据库体系带来了巨大的影响。这里所说的分布，主要是指以 WWW 全球网络为基础分布。Internet 网的迅速发展，网络上的资源日益丰富，

传统的那种固定模式的数据库形式已经显得力不从心。多媒体数据库系统将来肯定要考虑如何从 WWW 网络信息空间中寻找信息, 查询所要的数据。

(6) 传统的事务一般都是短小精悍, 在多媒体数据库管理系统中也应尽可能采用短事务。但有些场合, 短事务不能满足需要, 如从动态视频库中提取并播放一部数字化影片, 往往需要长达几个小时的时间, 作为良好的 DBMS 应保证播放过程不致中断, 因此不得不增加处理长事务的能力。

(7) 服务质量的要求。许多应用对多媒体数据的传输、表现和存储的质量要求是不一样的, 系统所能提供的资源也要根据系统运行的情况进行控制。对每一类多媒体数据都必须考虑这些问题: 如何按所要求的形式及时地、逼真地表现数据? 当系统不能满足全部的服务要求时, 如何合理地降低服务质量? 能否插入和预测一些数据? 能否拒绝新的服务请求或撤消旧的请求?

(8) 多媒体数据管理还有考虑版本控制的问题。在具体的应用中, 往往涉及对某个处理对象(如一个 CAD 设计或一份多媒体文献)的不同版本的记录和处理。版本包括两种概念。一是历史版本, 同一个处理对象在不同的时间有不同的内容, 如 CAD 设计图样, 有草图和正式图之分; 二是选择版本, 同一处理对象有不同的表述或处理, 一份合同文献便可以包含英文和中文两种版本。需解决多版本的标识和存储、更新和查询, 尽可能减少各版本所占存储空间, 而且控制版本访问权限。现有通用型 DBMS 大都没有提供这种功能, 而由应用程序编制版本控制程序显然是不合适的。

### 3. 多媒体数据的管理

应用程序开发者和数据库管理者面临的最大挑战是要把不同形式的信息, 包括文本、图像和视频组合在他们的应用程序中。正如我们在前面看到的那样, 即使是压缩了的多媒体对象容量也是非常大的。另外, 目前大多数多媒体应用都与网络通信相结合, 如电子邮件, 因此数据库系统必须是完全分布式的。现在有若干数据库存储方法可供选择, 这些方法将决定整个方案的灵活性和性能。可选的方法如下:

(1) 对现有关系数据库管理系统(RDBMS)进行扩展, 用二进制对象的方式支持各种多媒体对象。

(2) 把关系数据库中基本二进制对象扩展为继承和类的概念。支持这些特性的数据库管理系统提供对象程序设计前端扩展或 C++ 支持。

(3) 将数据库和应用程序转换为面向对象的数据库, 并使用 C++ 或 SQL 这样的面向对象的语言进行开发。

关系数据库是当前占主流地位的数据库, 但它缺乏对多媒体数据库的支持, 因为后者是数字和文本数据、GUI 前端的图像、CAD/CAM 系统和 GIS 应用程序、视频、音频和记录有音乐、伴音的全动作视频的组合。在关系数据库上实现多媒体应用程序的关键局限是其关系数据模型和关系计算模型。

关系数据库在设计上就仅支持表结构的字母数字数据(包含用二进制方式存储的日期), 绝大多数关系数据库支持整型、浮点、字符串、货币、日期、布尔量以及其他一些数据类型, 有一些也增加了一些新特点, 如在大二制对象 BLOB(Binary Large Object)中查询超文本, 但它们不支持如派生和聚合这样的类关系。另外, 未加修改的关系模型不支持自动管理数据的同步编辑, 如版本管理。关系数据库的计算模型也不支持扩展结构的遍历操作所需要的内

存驻留对象的概念。例如，在演示一个由包含有图像和全动作视频剪辑的 RTF 文本时的操作。而且，关系模型不能进行具有某种复杂度的长周期事务处理，像需要对被多个用户访问的分布式多媒体对象进行更新操作。

(1) RDBMS 的多媒体扩展。大多数先进的关系数据库将 BLOB 作为新的数据类型看作二进制和自由格式文本。BLOB 构成关系表中的列，用于图像和其他的二进制数据类型。关系数据表包含 BLOB 的位置信息，而 BLOB 实际存于数据库外部的独立的图像、视频服务器中。关系数据库经过扩充能访问这些 BLOB，从而提供给用户一个完整的数据集。例如：RDBMS 可能支持传统图像服务器（如有多个盘片的光盘库）上图像的 BLOB 的存储，但对 BLOB 的使用和操作也不是没有问题，如索引的设计。图像管理系统的设计（也可以用到多媒体系统中）将文件名作为 RDBMS 表，这个域说明图像（声音和视频段）的位置（服务器、卷号、文件名），图像或视频管理系统将索引域作为指向图像和视频文件的指针（关键字）以便读取所需对象。正如人们的想像，数据的读取需要一系列操作：从 RDBMS 中读取索引键传递给多媒体对象存储系统，该系统读取适当的对象，并传回给应用程序，应用程序把对象和相关的数据库信息显示出来。尽管属性数据和对象在不同的网络服务器上，这些工作必须紧密地配合。RDBMS 无法理解 BLOB 的内容，不能在 BLOB 内部进行索引。BLOB 可以有 2GB 或更大，这对大多数多媒体对象已经够了。

扩充的关系数据库正逐步向面向对象的系统发展。关系数据库具有进行严格的集合管理的能力，从而保证了数据库的完整性，而早期的 DBMS 中正是缺少这种完整性。RDBMS 比 ODBMS 多出的两个重要特点是安全性和事务的完整性。安全性是以对域级、表级和数据库级的访问形式提供的，多媒体数据事实上存放在数据库系统外部，因而无法获得数据库对其自身数据提供的安全保护。许多数据库外部的应用程序要不断对文件中存储的图像、声音信息或全动作视频进行访问。RDBMS 的事务完整性是通过保证当事务更新失败时事务被退回以保持数据库中受影响的部分的同步性。由于复合文件中对所有多媒体部件的访问要进行很多操作，所以事务的完整性的实现是很复杂的。事务的许多活动是不包含在数据库的事务控制中的，如读取多媒体部件、将其集成到可演示的对象中、记录其变化，然后将更新写回存储器。例如，如果不进行特定的操作来完成更新，RDBMS 是无法知道这一段视频被改变了的。

RDBMS 模型的另一个有意思的扩展是，除了存储在数据库内部以外，BLOB 也受到并发性和事务控制的影响。用户可以对 BLOB 有完全的事务控制进行读取和更新，也就是说，如果事务的后续部分失败了，对 BLOB 进行的所有修改都可以被退回。并发控制对 BLOB 的作用与对它数据类型的作用是一样的。

在基于 RDBMS 的客户-服务器应用程序中用得最多的过程化语言——标准 SQL 语言中，没有对数据库结构中的 BLOB 进行处理的协议。虽然许多独立的应用程序都可以操作 RTF（有图形和其他数据类型嵌入其中的文本）、图形、声音、全动作视频和其他数据，但这些信息只能保存在数据库结构之外。为了克服 SQL 在多媒体对象处理方面的缺点，可以将基于 SQL 的非过程性语言和单纯面向对象的数据库合并起来，使得多媒体对象的控制更加方便灵活。结果是数据库可以支持封装（将软件实体作为单元来处理）和继承（从已存在的类派生出新的类）这两个面向对象概念中的基本原则。封装是使用嵌套的表格来提供类似关系数据库的支持。关系数据库模型包括一系列关系，称为表格，它由行列组成。表格中的

一行表示单个关系，表格嵌套就是一个表格的一列中包含另一表格的所有行的内容。另外，表格允许有对每一行中某一列内容进行处理的过程。这样定义的关系表格就将数据库结构及行上的操作有效地封装起来。任何过程都可以与表格相联系并对表格中的每一行进行处理。这样，面向对象数据管理的两个先决条件——继承和封装，就已经具备了，并且整个是在关系数据库模式内部进行。

关系数据库提供的数据库访问的灵活性是多媒体文档管理系统的重要特征。复合关键字使得对多媒体文档的访问可以有多种组合方式，这方面数据库提供了完全的适应性。但是，前面提到的显著缺点和面向对象系统在这些领域的优点是使用面向对象程序设计方法的最有力的实证。

(2) 多媒体的面向对象数据库。尽管有扩展的关系数据库，对象数据库（其中数据仍然在关系数据库或一般文件之中）是对多媒体的支持的最快捷的途径。运用可复用代码和模板概念的面向对象编程使得数据库的维护更简单。但是它会带来危险。当前的对象数据库缺乏安全性和并发控制，使得目前大多数对象数据库无法用于商业应用。但是类的概念和面向对象数据库模型非常适合多媒体数据。只要建立了类，其中所有的对象都有了该类的属性。类的定义能加速应用程序的开发速度，还能提供更广泛的对象能力和对复杂的多媒体应用的开发和维护工具。对象数据库的诸如消息传送、可扩展性和对层次结构的支持等能力对多媒体系统是很重要的。

ODMS（对象数据库）允许对数据库应用程序的增量修改，这些修改在过程化语言环境下会更困难。例如，“消息传送”允许对象之间以激发各自方法的方式相互作用，应用程序的一个成员的数据到另一个成员的传送过程能导致使数据的管理更适合下一层次。数据库中这种方式的分层支持使得设计更为简单，因为大多数产品设计都包容了客户与服务器之间的自然的分层结构。“可扩展性”意味着操作、结构、约束的集合都不是固定的，正如 RDBMS 中那样，开发者可为他们的应用程序增加新的操作。早期面向对象数据库缺乏关系数据库的鲁棒性，其中的事务完整性仍然难以控制，像恢复、撤消这些常用特性还没有被完全支持。尽管如此，面向对象数据库仍然为多媒体软件提供了新的有力的基础，下面我们来重温一下面向对象数据库的关键概念以理解它为什么对多媒体系统如此重要。

面向对象的软件技术基于下面三个概念，对多媒体系统是十分重要的：

- 封装性，或者说以预定义、可控制的方式把软件实体作为单元来处理，其中控制程序是与实体结合在一起的。
- 联系，或以与另一实体的差异来定义一个软件实体的能力。
- 分类，或以有相同行为、属性的数据项来代表一个单独的软件实体的能力。

面向对象的一个重要优势是以模块化和复用方式来组织软件。领域独立的类库定义了属性和控制程序的基本集，这个基本集是一种复杂的信息元素类型——对象所要求的。例如，对象可能是一个视频片断或一特别复杂的文档。

面向对象的另一个重要性是提供了较软件实体更直接的模拟复杂的现实世界的新方法。领域相关的类库被用来定义真实世界的属性而不是这些实体的抽象属性，这些属性在类的下一层定义。

类库还被用来支持诸如数据转换、符合用户环境的数据表示这样的函数。专用针对交互领域的类库简化了处理大量显示系统、压缩/解压缩技术的需要。

封装的另一个好处是，它允许真正开放系统的开发，在其中应用程序的一部分不需要知道另一部分的功能。封装成功地隐藏了内部成员的功能，只留下了交互接口。一个面向对象程序允许程序员直接表达封装、聚合、分类的概念。对开放系统，一个类库有可能为其他遵循相同运行规则的程序提供开放接口。

对开放系统，对象的接口（当从外部观察时它的公共特征）和实现（产生这些外观特征的私有属性和函数）之间的差别很重要。封装提供自治性，就是说，对各种外部程序的接口能在一类对象上实现而数据存储在其他类对象中。一个对象不能影响其他对象的运作，只限于使用其他对象的公共方法。

继承机制允许快速构造与父辈性质相近的对象，如从一基本显示对象继承得到一系列不同的显示对象来适应工作站的需要。

开放系统的用户能从为网络中不同系统建立的面向对象机制的基本标准中获得极大好处，该标准允许互相通信，提供在此之上的服务。Object Management Group 的对象请求代理（object request broker）就试图建立这样一个通信机制。例如，MS Windows、Windows NT、IBM Presentation Manager、NextStep、X-Window 和 Sun 的 Open Look 等都有具有公共特性的面向对象的用户接口，但它们的结构是很不一样的，MS Windows 提供大量非面向对象的 API 函数，Sun 的 Open Windows 提供 X Window 图形 API 函数，一个非面向对象的过程化的接口。Next Step 提供了详尽的类库，但基本图形功能通过 Display PostScript System 提供，NEWS System 建立在 PostScript 语言上再加上覆盖整个用户界面的面向对象结构，X Window 系统依靠工具包兼容性，它建立在类机制上（通过由编程语言以外的软件惯例库来实现的继承性）。对象请求代理可用来定义功能，而实际的操作由理解 API 的本地对象实现。以这种方式，对象请求代理只需要知道对 GUI 环境的公共接口，而不是具体编程机制。

### 8.1.3 多媒体数据库体系结构

由于目前还没有完好的多媒体数据模型，所以还没有标准的多媒体数据库体系结构，现在大多数解决办法是采用扩展现有的关系数据库，一是扩展字段长度，一是扩展为对象，采用面向对象的方案。这里介绍的仅局限在专门的应用范围，只对专门应用进行结构设计。

#### 1. 组合型多媒体数据库结构

针对各种媒体单独建立数据库，每一种媒体的数据库都有自己独立的数据库管理系统。虽然它们是相互独立的，但可以通过相互通信来进行协调和执行相应的操作。用户既可以对单一的媒体数据库进行访问，也可以对多个媒体数据库进行访问以达到对多媒体数据进行存取的目的。这种多媒体数据库系统的体系结构示意图如图 8-1 所示。在这种数据库体系结构中，对多媒体数据的管理是分开进行的，可以利用现有的研究成果直接进行组装，每一种媒体数据库的设计也不必考虑与其他媒体的匹配和协调。但是，由于这种多媒体数据库对多媒体的联合操作实际上是交给用户去完成的，给用户带来灵活性的同时，也为用户增加了负担。该体系结构对多种媒体的联合操作、合成处理和概念查询等都比较难于实现。如果各种媒体数据库设计时没有按照标准化的原则进行，它们之间的通信和使用都会产生问题。

## 2. 集中型多媒体数据库结构

只存在一个单一的多媒体数据库和单一的多媒体数据库管理系统，各种媒体被统一地建模，对各种媒体的管理与操纵被集中到一个数据库管理系统之中；各种用户的需求被统一到一个多媒体用户接口上，多媒体的查询检索结果可以统一地表现。由于这种多媒体管理系统是统一设计和研制的，所以在理论上能够充分地做到对多媒体数据进行有效的管理和使用。但实际上这种多媒体数据库系统是很难实现的；目前还没有一个比较恰当而且效率很高的方法来管理所有的多媒体数据。虽然面向对象的方法为建立这样的系统带来了一线曙光，但要真正做到还有相当长的距离。如果把问题再放大到计算机网络上，这个问题就会更加复杂。结构示意图如图 8-2 所示。

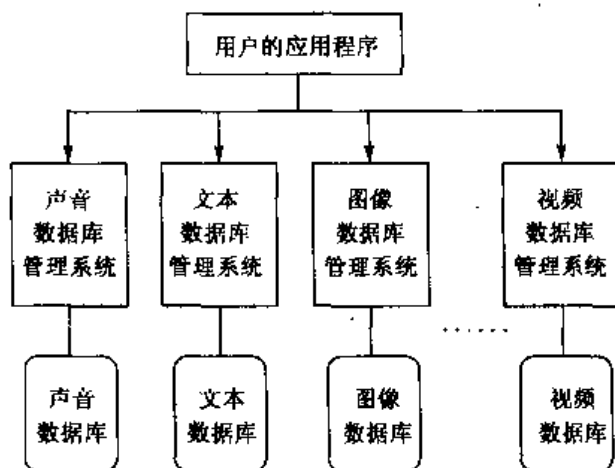


图 8-1 组合型多媒体数据库结构

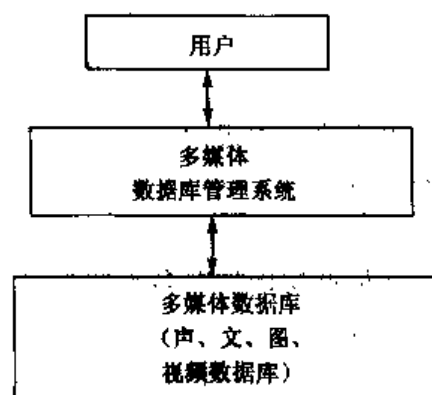


图 8-2 集中型多媒体数据库结构

## 3. 客户 / 服务器型多媒体数据库结构

减少集中型多媒体数据库系统复杂性的一个很有效的办法是采用客户 / 服务器结构。各种单媒体数据仍然相对独立，系统将每一种媒体的管理与操纵各用一个服务器来实现，所有服务器的综合和操纵也用一个服务器完成，与用户的接口采用客户进程实现。客户与服务器之间通过特定的中间系统连接。使用这种类型的体系结构，设计者可以针对不同的需求采用不同的服务器、客户进程组合，所以很容易符合应用的需要，对每一种媒体也可以采用与这种媒体相适合的处理方法。同时，这种体系结构也很容易扩展到网络环境下工作。但采用这种体系结构必须要对服务器和客户进行仔细的规划和统一的考虑，采用标准化的和开放的接口界面，否则也会遇到与联组合相近的问题。该体系结构的示意图如图 8-3 所示。

## 4. 多媒体数据的层次结构

传统的数据库系统分为三个层次，按 ANSI 的定义分别为物理模式、概念模式和外部模式。传统的数据库采用这种层次结构是因它所管理的数据而决定的。在这种数据库中，数据主要是抽象化的字符和数值，管理和操纵的技术也是简单的比较、排序、查找和增删改等操作，处理起来容易，也比较好管理。由于数据种类单一，数据模型比较简单，对数据的处理也可以采取相对统一的方法，用户除了表格之外没有更复杂的数据表现工作。因此，如果要引入多媒体的数据，这种系统分层肯定不能满足要求，就必须寻找恰当的结构分层形式。



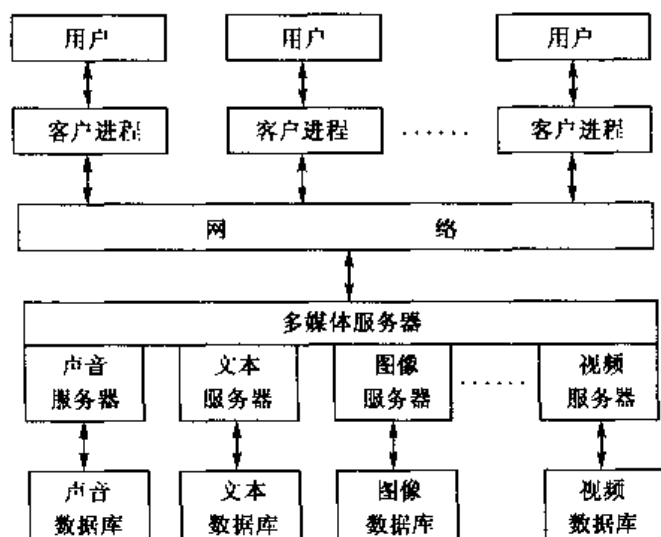


图 8-3 客户/服务器型多媒体数据库结构

已经有许多人提出过多媒体数据库的层次划分，包括对传统数据库的扩展、对面向对象数据库的扩展、超媒体层次扩展等。虽然各有所不同，但总的思路是很相近的，人都是从最低层增加对多媒体数据的控制与支持，在最高层支持多媒体的综合表现和用户的查询描述，在中间增加对多媒体数据的关联和超链的处理。在这里，综合各种多媒体数据库的层次结构的合理成分，我们提出一种多媒体数据库层次划分的概念结构图，如图 8-4 所示。

在该图中，最低层也就是第一层，称为媒体支持层，建立在多媒体操作系统之上。针对各种媒体的特殊性质，在该层中要对媒体进行相应的分割、识别、变换等操作，

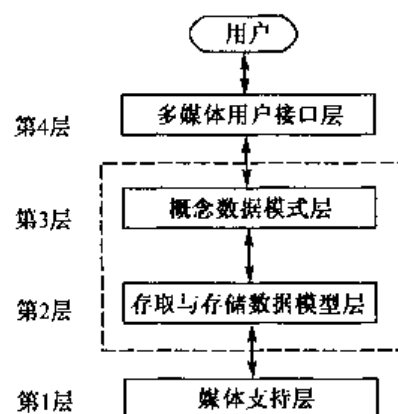


图 8-4 多媒体数据库的层次示意图

并确定物理存储的位置和方法，以实现对各种媒体的最基本数据的管理和操纵。由于媒体的性质差别很大，对于媒体的支持一般都分别对待，在操作系统的辅助下对不同的媒体实施不同的处理，完成数据库的基本操作。第二层称为存取与存储数据模型层，完成多媒体数据的逻辑存储与存取。在该层中，各种媒体数据的逻辑位置安排、相互的内容关联、特征与数据的关系以及超链的建立等都需要通过合适的存取与存储数据模型进行描述。第三层称为概念数据模型层，是对现实世界用多媒体数据信息进行的描述，也是多媒体数据库中在全局概念下的一个整体视图。在该层中，通过概念数据模型为上层的用户接口、下层的多媒体数据存储和存取建立起一个在逻辑上统一的通道。第三和第二层也可以通称为数据模型层。第四层称为多媒体用户接口层，完成用户对多媒体信息的查询描述和得到多媒体信息的查询结果。很显然，这一层在传统数据库中是非常简单的，但在多媒体数据库中这一层却成了最重要的环节之一。用户首先要能够把他的思想通过恰当的方法描述出来，并能使多媒体系统所接受，这在多媒体数据库系统中本身就是一个十分困难的问题，不是用某一种类似于 SQL 之类的语言所能描述的。次之，查询和检索到的结果需要按用户的需求进行多媒体化的表现，甚至构造出“叙事”效果，这也是表格一类接口所不能做到的。

上面的多媒体数据库的层次划分当然是非常概念化的，也是很初步的。多媒体数据库的



结构应该能够包含像图像数据库、视频数据库、全文数据库等一系列的专业数据库类型，并能统一地管理和使用，但目前离这一目标还很远。

## 8.2 多媒体数据库基于内容检索

随着多媒体技术的迅速普及，多媒体数据库的应用将越来越广泛，而每一种媒体数据都具有难以用符号化的方法描述的信息线索，例如，图像中的颜色、对象分布，视频中的运动、事件，音频中的音调等。当用户希望利用这些信息线索对数据进行检索时，传统的基于关键词的数据库检索方式不再适用。其主要原因有两个方面：一方面，在许多情况下多媒体内容难以用仅仅几个关键词来充分描述，而且作为关键词的媒体特征的选取也有很大的主观性，另一方面，用户很难将这些信息线索转化为某种符号的形式。那么，如何从多媒体数据库中检索出符合条件的多媒体信息？为此，基于内容的检索（content-based retrieval）技术诞生了。所谓基于内容的检索，就是从媒体数据中提取出特定的信息线索，然后根据这些线索从大量存储在数据库中的媒体中进行查找，检索出具有相似特征的媒体数据。基于内容的检索应具有如下特点：

- 从媒体内容中提取信息线索。基于内容的检索突破了传统的基于关键词检索的局限，直接对图像、视频、音频进行分析，抽取特征，使得检索更加接近媒体对象。
- 基于内容的检索是一种近似匹配。在检索过程中，采用逐步求精的办法，每一层的中间结果是一个集合，不断减小集合的范围，直到定位到目标。这一点与数据库检索的精确匹配算法有明显不同。

基于内容的检索可以利用图像处理、语音信号处理、模式识别、计算机视觉等学科中的一些方法作为部分基础技术。因为基于内容的检索不仅仅是基于内容，而且从应用的关键技术上看，其本质上是一门信息检索技术。它利用认知科学、用户模型、图像处理、模式识别、知识库系统、计算机图形学、数据库管理系统、信息检索等领域的研究成果和方法，研究新的媒体数据的表示和数据模型、有效和可靠的查询处理算法、智能查询接口以及与应用领域无关的系统结构。基于内容的检索与模式识别、图像理解、计算机视觉等学科的重要区别是，基于内容的检索是一种信息检索技术，要能够从大型分布式数据库中以用户可以接受的响应时间查询到要求的信息，它不一定需要去理解和识别媒体中的目标，关注的是以基于内容或特征的方法快速发现信息。

### 8.2.1 基于内容检索系统的体系结构

#### 1. 基于内容检索的体系结构

基于内容的检索作为一种信息检索技术，接入或嵌入到其他多媒体系统中，如超媒体（浏览器）系统、会议系统、多媒体信息系统、关系数据库系统等，提供基于多媒体数据内容的信息查询和检索。因此，将基于内容的检索设计为多媒体数据库的检索引擎结构，在体系结构上划分为两个子系统：特征抽取子系统和查询子系统，如图 8-5 所示。

各个模块的主要功能是：

（1）目标标识。目标标识为用户提供一种工具，以全自动或半自动（需要用户干预）的方式标识静态图像、视频镜头的代表帧等媒体中用户感兴趣的区域，以及视频序列中的动

态目标，以便针对目标进行特征提取并查询。当进行整体内容检索时，利用全局特征，这时不用目标标识功能。目标标识是可选的。

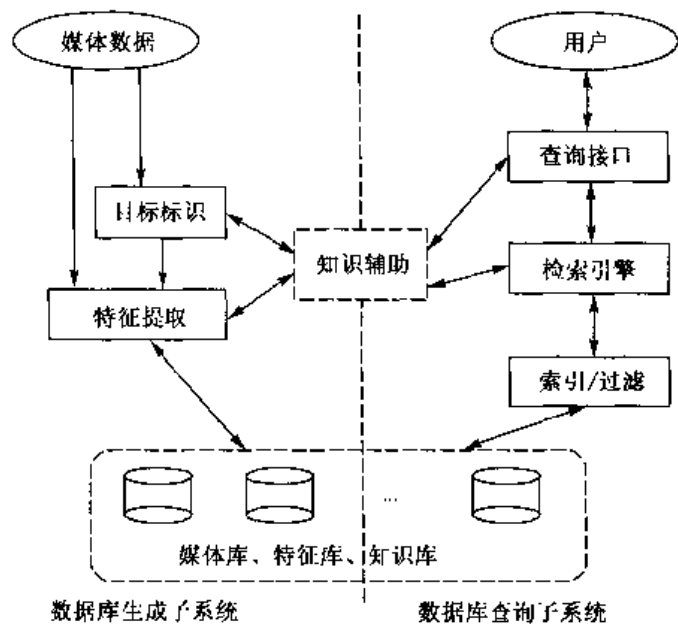


图 8-5 基于内容检索的体系结构

（2）特征提取。对多媒体数据进行特征提取，提取用户感兴趣的、适合检索要求的特征。特征提取可以是全局性的，如整幅图像的视频镜头，也可以是针对某个目标的，如图像中的子区域，视频中的镜头和运动对象等。

（3）数据库。生成的数据库由媒体库（集）、特征库（文件）和知识库组成。媒体库包含多媒体数据，如图像、视频、音频、文本等。特征库包含用户输入的特征和预处理自动提取的内容特征。知识库中知识表达可以更换以使用各种不同的应用领域。

（4）查询接口。友好的人机交互界面是一个成功检索系统不可缺少的条件，它可以大大提高检索的效率。在基于内容检索中，由于特征值为高维向量，不具有直观性，因此必须为其提供一个可视化的输入手段。可采用的方式有三种：操纵交互输入方式、模板选择输入方式和用户提交特征样板的输入方式。同时应支持多种特征的组合。另外，查询返回的结果需要浏览，应在用户界面提供浏览功能。

（5）检索引擎。检索是利用特征之间的距离函数来进行相似性检索。模仿人类的认知过程，近似得到数据库的认知排队，对于不同类型的媒体数据有各自不同的相似性测度算法，检索引擎中包括一个较为有效可靠的相似性测度函数集。

（6）索引/过滤器。检索引擎通过索引/过滤模块达到快速搜索的目的，从而可以应用到数据库中的大型多媒体数据集中。过滤器作用于全部数据，过滤出的数据集合再用高维特征匹配来检索。索引用于低维特征，可以用 R\*树索引来加快检索。

## 2. 基于内容检索的处理过程

基于内容的查询和检索是一个逐步求精的过程，存在着一个特征调整，重新匹配的循环过程。如图 8-6 所示。

（1）提交查询要求。用户查找一个数据对象时，利用系统人机界面提供的输入方式（可

视觉化的输入界面或查询语言) 形成一个查询条件。与传统的文本查询不同的是, 在将查询条件传递给搜索引擎之前, 一般要对所提交的数据进行预处理, 在分布式应用中这一点尤为重要。

(2) 相似性匹配。将查询特征与数据库中的特征按照一定的匹配算法进行匹配。

(3) 返回候选结果。满足一定相似性的一组候选结果按相似度大小排列返回给用户。

(4) 特征调整。对系统返回的一组初始特征的查询结果, 用户可以通过遍历(浏览)挑选到满意的结果, 也可以从候选结果中选择一个示例进行特征调整, 最后形成一个新的查询。

如此逐步缩小查询范围, 直到用户对查询结果满意为止。

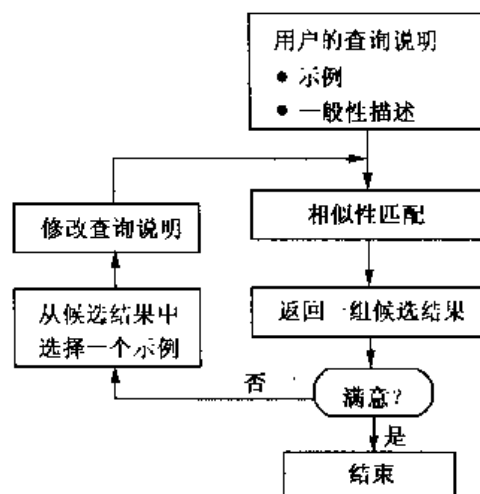


图 8-6 基于内容检索的处理过程

## 8.2.2 基于内容检索的关键技术

基于内容检索需要解决两类关键技术可以总结出以下两种:

- 多媒体特征提取和匹配。
- 相似检索技术。

### 1. 相似性匹配的一般方法

由于多媒体的内容语义无法十分精确, 因此采用相似性的匹配方法。下面以美国 Ramesh Jain 等人设计的视觉信息管理系统 (VIMS) 为例, 从下面 5 个方面介绍特征相似性匹配的一般方法。

(1) 模糊值。一般地, 模糊集与隶属函数可以定义如下: 所谓给定论域  $U$  上的一个模糊集  $A$  是指, 对任何  $u \in U$ , 都指定一个数  $f_A(u) \in [0, 1]$  与之对应,  $f_A(u)$  便称为  $u$  对  $A$  的隶属度。这意味着作出了一个映射:

$$f_{A_i} : U \rightarrow [0, 1]$$

$$u \rightarrow f_A(u)$$

这个映射称为  $A$  的隶属函数, 其中论域  $U$  是指被讨论对象的全体。

用隶属函数  $f_A$  描述特征的结果称为特征的模糊值。例如, 若描述一幅图像的颜色特征, 就可以用模糊值的概念,  $f(x, \text{blue}) = 0.72$  可表示是图像  $x$  等于“蓝色”的可能性是 72%。

(2) 特征值分类计算。VIMS 中有三个特征集合:  $F_u$ 、 $F_d$ 、 $F_c$ 。 $F_u$  是由用户描述的特征值, 在图像存入时指定, 如年龄、性别等客观值等, 以规范化形式存储;  $F_d$  是从图像中直接提取的特征, 存入数据库时由系统自动计算提取, 存储提取结果;  $F_c$  是直到需要时才计算的特征值, 不存储。

例如, 当查找一个面部对象时, 由用户指定一些  $F_u$  客观值, 如年龄、性别、面部轮廓(宽、窄、中等)、头发长度等; 对每一个指定的客观特征值需要通过一个全局统计算法映射成一个标准值, 对没有指定的值, 由系统参照已说明的其他特征自动赋给一个标准值。

(3) 特征调整。使用初始特征值查找出来的不是一幅而是一组图像。一种情况是从该组图像中能挑选出所需的图像; 另一种情况是挑不出来, 此时需挑选一幅图像进行特征调整。

例如头发再长一点，眼睛再大一点等等，调整的方法是：

$$f_i' = f_i + \Delta_u(i) \times \delta_i$$

其中： $f_i'$  是调整后的  $i$  的特征值， $f_i$  是  $i$  的当前值， $\delta_i$  是特征  $i$  的标准误差， $\Delta_u(i)$  是新的调整映射成特征的结果，计算方法是： $\Delta_u(i) = k \times \delta_i / N^{1/2}$ ， $k$  是被检索到的图像数量， $\delta_i$  是调整的大小， $N$  是数据库中含特征  $f_i$  的图像对象的个数。

(4) 相似特征偏差。为了检索出与目标图像相似的图像集合，需要计算各图像的特征之间的相似程度，即相差程度。例如计算两个面部图像  $F_1$  和  $F_2$  在特征  $f_i$  上相差程度的函数为：

$$\text{delta}(i, F_1, F_2) = (F_1(i) - F_2(i))^2 / \delta_i$$

(5) 相似度及最终求值。所谓“相似度”是指相似性的度量结果。相似度的度量首先在各个阶段上进行，再将每个阶段综合考虑求出最终结果。例如，对人头像的面部进行求值，首先分别度量两幅图像上的人的眼睛、鼻子、嘴等的相似度，不同阶段（部分）的比较对整个相似度的影响不一样，一般眼睛比鼻子更重要，因此需对不同的阶段分配不同的权重。用下面的公式描述面部偏差：

$$\text{Diff}(F_c, F_u) = \sum_i W_i \times \text{Seg}_i \times \text{Conf}_i \times \text{delta}(i, F_c, F_u) \quad (8-1)$$

其中：

$W_i$ ：特征的权重（重要程度），可以根据需要按同意尺度确定，如 0~10、0~1.0 均可。

$\text{Seg}_i$ ：对  $f_i$  分段的可信程度。

$\text{Conf}_i$ ：对  $f_i$  的值的可信程度。

$\text{Delta}(i, F_1, F_2)$ ：计算两个面部图像  $F_1$  和  $F_2$  在特征  $f_i$  上相差程度的函数：

$$\text{delta}(i, F_1, F_2) = \frac{(F_1(i) - F_2(i))^2}{\delta_i} \quad (8-2)$$

显然， $\text{Diff}(F_c, F_u) \geq 0$ 。当所有的  $F_c$  都与  $F_u$  进行比较后，会得到一组偏差值，例如：

$\text{Diff}(F_c, F_u) = \{9.66, 5.21, 11.08, 3.5, \dots\}$ ，此时可将具有最小偏差值的图像（一个或几个）返回给用户，或是确认，或是继续进一步的查询。

## 2. 图像的特征提取与匹配

(1) 颜色直方图原理。假设一幅图像  $G$  的颜色（或灰度）由  $N$  级组成，每一种颜色值用  $q_i$  ( $i=1, 2, \dots, N$ ) 表示。在整幅图像中，具有  $q_i$  颜色值的像素数为  $h_i$ ，则这一组像素统计值  $h_1, h_2, \dots, h_N$  就是该图像的颜色直方图，可用  $H(h_1, h_2, \dots, h_N)$  表示。

如果用  $H(h_1, h_2, \dots, h_N)$  来描述一幅图像的颜色特征，则该直方图具有如下性质：

1) 直方图中的值都是统计而来，描述了该图像关于颜色的数量特征，可以反映图像的部分内容。举例来说，一幅“蓝色的海洋”的图像，“蓝色”将是像素的主要成分，在数量上占很大的比例。

2) 直方图丢失了颜色的位置特征。因此，不同的图像可能具有相同的颜色分布，从而也就具有相同的颜色直方图。

3) 如果将图像划分为若干子区域，所有子区域的直方图之和等于全图直方图。

4) 一般情况下，由于图像上的背景和前景物体颜色分布明显不同，从而在直方图上会

出现双峰特性,但前景和背景颜色较为接近的图像不具备该性质。

(2) 颜色直方图的相似性匹配。假设示例图像直方图用  $G(g_1, g_2, \dots, g_N)$  表示,数据库中的目标图像直方图用  $S(s_1, s_2, \dots, s_N)$  表示,可以把这两个直方图看作欧氏空间的两个点,利用欧氏距离来描述它们的相似性,即:

$$Ed(G, S) = \sqrt{\sum_{i=1}^N (g_i - s_i)^2} \quad (8-3)$$

将上式规范化并简化,按“1”为完全相似,“0”为完全不相似来描述两个图像的相似性,则有:

$$Sim(G, S) = \frac{1}{N} \sum_{i=1}^N \left( 1 - \frac{|g_i - s_i|}{Max(g_i, s_i)} \right) \quad (8-4)$$

其中  $N$  为颜色级数,  $g_i \geq 0$ ,  $s_i \geq 0$ 。如果  $Sim$  靠近“1”,则认为两幅图像在颜色上完全相似;否则,不相似。

如果对某些颜色要有重要程度的区别,可以用权重因子  $W_j$  ( $0 \leq W_j \leq 1$ ,  $j=1, 2, \dots, N$ ) 来描述,相似程度描述为

$$Sim(G, S) = \frac{1}{N} \sum_{i=1}^N \left\{ W_j \cdot \left[ 1 - \frac{|g_i - s_i|}{Max(g_i, s_i)} \right] \right\} \quad (8-5)$$

由于  $0 \leq W_j \leq 1$ , 它的引入导致  $Sim(E, S)$  的值下降,没有反映直方图中重要成分在相似性中的地位。为此,将其作一调整,从  $N$  个颜色值中选取  $L$  个最大的单元值进行求和平均,即:

$$Sim(E, S) = \frac{1}{L} \sum_{k=1}^L W_k \left[ 1 - \frac{(e_k - s_k)}{Max(e_k, s_k)} \right] \quad (8-6)$$

利用这个权重公式和直方图性质 4) 的双峰特性,结合相似性方法,可以确定重要特征或特征的组合。例如,可以做“寻找某一背景”、“寻找某一前景”、“A 图像的背景, B 图像的前景之组合的图像”等查询。

(3) 图像通用特征的描述。图像的通用特征有主颜色、纹理、颜色直方图、颜色分布、轮廓等。下面结合 *ImgRetr*——一个以 Internet/浏览器为平台的多媒体图像素材库基于内容检索系统的实现,说明上述通用特征的描述方法。

1) 主颜色。主颜色就是图片中占较大比重的颜色。主颜色的提取方法是对图像先计算颜色直方图再进行图像点的聚类。在 RGB 颜色空间对图像进行直方图计算,将 RGB 三个颜色都量化为 16 级,直方图的大小为  $16 \times 16 \times 16$ ,也就是有  $16 \times 16 \times 16$  种颜色。图像在点  $(x, y)$  处的 RGB 颜色响应分别为  $R(x, y)$ 、 $G(x, y)$ 、 $B(x, y)$ ,那么,颜色直方图为:

$$H_{RGB}(i, j, k) = \sum_{x, y} C(R(x, y), i) \cdot C(G(x, y), j) \cdot C(B(x, y), k) \quad (8-7)$$

其中,  $C(r, i)$  为表征值  $r$  是否属于盒子  $i$  的从属函数定义为:

$$C(r,i) = \begin{cases} 1 & 16 \times i \leq r < 16 \times (i+1) \\ 0 & \text{其余} \end{cases} \quad (8-8)$$

在得到颜色直方图后，利用 k-均值聚类算法可计算得到图像的主要颜色。若取 k 为 4，将含有像素数目超过像素总数 20% 的聚类中心作为该图的主颜色。显然，一幅图可以有多个主颜色。

2) 纹理。一些图像在局部区域内呈现不规则性，但在整体上表现出规律性，习惯上把图像这种局部的不规则而宏观有规律的特性称为纹理。纹理特征包括粗糙性、方向性和对比度。纹理主要用于布匹、墙纸等具有简单、重复性的图案的检索。纹理的描述采用 5 个方向、6 个尺度（2 倍递增）的 Gabor 系数，可采用如下方法计算：将图像通过尺度变换变换为标准大小（128×128）的灰度图像，记  $m$  方向  $n$  尺度的 Gabor 模板为  $g_{mn}(x, y)$ ，则图像的 Gabor 变换为：

$$W_{mn}(x, y) = \int I(x_1, y_1) g^*(x - x_1, y - y_1) dx_1 dy_1 \quad (8-9)$$

取  $V_{mn} = \max_{x, y} (|W_{mn}(x, y)|)$  作为此方向和尺度下的向量值，则纹理描述是一个  $5 \times 6 = 30$  维向量。

3) 轮廓。在对图像进行基于内容检索时，根据图像中物体的轮廓特征进行检索。目前较好的方法是采用图像的自动分割方法结合识别目标的前景和背景模型来得到比较精确的轮廓。由于图像轮廓是图像目标的大体描述，可用一些轮廓的简化特征作为检索的依据。一般以轮廓的中心为基准，计算中心到边界点的最长轴和最短轴、长轴与短轴之比、周长与面积之比、拐点等作为轮廓的检索特征。

下面介绍一种计算量较小的轮廓特征计算方法。可根据图像一般由大块颜色区域构成的特点，采用如下方法：首先将图像通过尺度变换使其成为标准大小（16×16 或 32×32 或 64×64），再计算图像颜色直方图并对其聚类。根据聚类结果对图像进行分割。对分割得到的图像采用类似中值滤波的方法去除小块区域和毛刺噪声，从而得到图像的大块区域颜色分割图。不同区域间的边界就构成了图像的轮廓。

由于不同图像产生的轮廓具有不同数目的轮廓点（轮廓的长短和数量不同），为便于比较对其进行归一化，从而给出轮廓特征向量的定义。

轮廓特征向量的定义：对  $L \times L$  大小的轮廓图像  $I_f$ ，其对应的轮廓特征向量  $v$  为一  $L \times L$  维的向量，该向量在第  $(i, j)$  维上的值定义为

$$v(i, j) = \frac{I_f(i, j)}{\sum_{m=0}^L \sum_{n=0}^L I_f(m, n)} \quad (8-10)$$

即轮廓特征向量是轮廓图像对其中轮廓点数目的归一化。从人的感知可知，轮廓间的相似度与轮廓的相互位置有关。欧氏距离已不能充分描述轮廓特征间的相似度计算。一种能充分考虑了各维坐标相关性的距离是二次型距离，其定义为：向量  $x$  和  $y$  间的二次型距离定义

为  $d_{quadric}^2(x, y) = (x - y)^t A (x - y)$ ，其中  $A = [a_{ij}]$  为二次距离的系数矩阵，元素  $a_{ij}$  是代表第  $i$  维和第  $j$  维间相似性的相关系数。对轮廓特征来说，采用  $v(i, j)$  形式来描述矩阵特征向量更为直观和方便，有

$$d_{sketch}^2(x, y) = \sum_{i=0}^L \sum_{j=0}^L \sum_{m=0}^L \sum_{n=0}^L a(i, j, m, n) [x(i, j) - y(i, j)] [x(m, n) - y(m, n)] \quad (8-11)$$

相关系数  $a(i, j, m, n)$  应仅与  $(i, j)$  和  $(m, n)$  间的关系有关，即存在一个函数  $f(i, j)$  满足  $a(i, j, m, n) = f(i - m, j - n)$ ，我们称此  $f(i, j)$  为距离相关函数，则轮廓特征间的距离可进一步写为：

$$d_{sketch}^2(x, y) = \sum_{i=0}^L \sum_{j=0}^L \sum_{m=0}^L \sum_{n=0}^L f(i - m, j - n) [x(i, j) - y(i, j)] [x(m, n) - y(m, n)] \quad (8-12)$$

由于轮廓特征维数较高，直接利用上式计算距离的计算代价很大。为了提高系统的实时性，需要对其进行改进。一般地，可用一个仅在  $(0, 0)$  附近小块区域上非零的  $g(i, j)$  来近似  $f(i, j)$  进行轮廓特征间距离的计算。考虑对称性和各向同性，应有  $g(i, j) = g(-i, j) = g(i, -j) = g(-i, -j)$ ，且  $g(i, j) = g(j, i)$ 。亦可知  $(i, j)$  非零区域的大小为奇数，记为  $2r + 1$ 。则除边缘处，公式 (8-12) 可写为：

$$d_{sketch}^2(x, y) = \sum_{i=0}^L \sum_{j=0}^L [x(i, j) - y(i, j)] \sum_{m=-r}^r \sum_{n=-r}^r g(m, n) [x(i - m, j - n) - y(i - m, j - n)]$$

进一步可写为：

$$d_{sketch}^2(x, y) = \sum_{i=0}^L \sum_{j=0}^L [x(i, j) - y(i, j)] [g(i, j) * [x(i, j) - y(i, j)]] \quad (8-13)$$

其中 “\*” 为卷积符号。因此，轮廓特征间的距离可通过特征差和特征差与一小模板的卷积的内积来加以计算。

### 3. 视频特征的提取与匹配

与图像检索相比较，视频检索含有更丰富的内容。但与图像数据一样，原始的视频数据也是非结构化的，在检索前要对其结构化。为此，首要工作是将此连续的视频流划分为具有特定语义的视频片段来作为检索的基本单元，在此基础上才能作进一步的描述和组织。这样，视频检索就包括视频分割和描述建立两部分内容。

#### (1) 视频分割：

1) 视频序列的组成。根据视频信号的产生原理，我们知道一般的视频序列本身是由多段镜头（摄像机上从记录 RECORD 到停止 STOP 期间所获得的一段连续视频信号）通过拼接编辑而成，而每个镜头本身具有一定的独立性和完整性。每一个镜头包含一个事件或一组连续的动作。每个镜头中的内容发生在一个场景（scene）中，一个场景可以分散在多个镜头之中。因此，在实际应用中多是以视频流中的自然分界——镜头场景的切换作为视频流的分割依据。视频分割即将连续的视频流分割为相对独立的视频片段。根据应用背景的不同，对视频流的分割可以采用不同的标准，比如也可以把某一物体在图像中的出现和消失定义为片段的头和尾。从这个意义上说，视频分割即对视频流中帧图像内容连续性

中断的检测。

2) 镜头的切换。镜头的切换点是视频序列中两个不同镜头之间的分隔和衔接, 是在导演切换台上或特技发生器上做出来的。切换的方法主要有直接切换和渐变切换两类。

一个镜头与另一个镜头之间没有过渡, 由一个镜头的瞬间直接转换为另一个镜头的方法叫做直接切换。由于画面的改变是在视频的消隐场期间进行的, 所以画面的接点不会出现跳动。镜头切换是最简单的视频编辑, 通过将两个镜头直接拼接形成, 镜头转换间不做任何编辑。其模型如图 8-7 所示。镜头  $A(t)$  和镜头  $B(t)$  的长度分别为  $t_1$  和  $t_2$ , 由镜头  $A(t)$  切换到镜头  $B(t)$ , 形成一个长度为  $t_1+t_2$  的镜头  $C(t)$ ,  $C(t)$  定义为

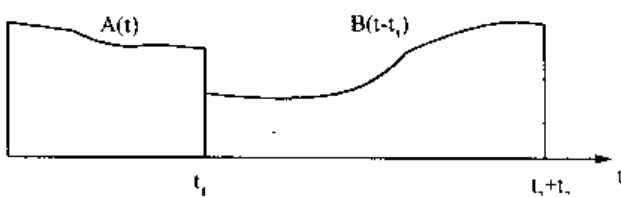
$$C(t) = \begin{cases} A(t) & 0 \leq t < t_1 \\ B(t-t_1) & t_1 \leq t < t_1+t_2 \end{cases}$$


图 8-7 镜头切换模型

镜头渐变是指镜头与镜头之间的变换是缓慢过渡的, 没有明显的镜头跳跃。包括淡入 (fade in)、淡出 (fade out), 慢转换 (diss)、扫换 (wipe) 等。将画面逐渐关闭消失称为淡出, 将画面逐渐加强称为淡入。一个画面消失的同时另一个画面逐渐出现称为慢转换。图像从画面的某一部分开始逐渐地被另一画面取而代之的方式称为扫换。扫换是由特技发生器产生出来的, 方式有上百种。这些镜头切换的技巧使得镜头之间的连接更加紧密。镜头渐变是一段视频镜头在镜头转换处在一段时间内渐渐转变为后续视频镜头的编辑方式。在实际视频信号编辑中, 镜头渐变一般通过线性特性的光学组件或相应的视频编辑设备来完成。镜头渐变的信号构成方式如图 8-8 所示。

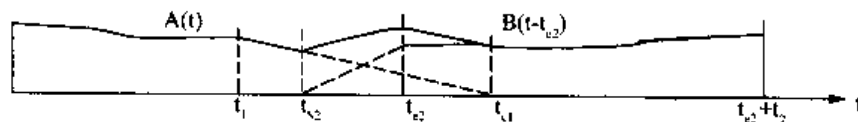


图 8-8 镜头渐变模型

图中, 在长度为  $t_1$  的镜头  $A(t)$  和长度为  $t_2$  的镜头  $B(t)$  间存在着一个镜头渐变过程。镜头  $A(t)$  在  $t_1$  处终止, 经过  $t_{e1}-t_1$  长的线性衰减后, 在  $t_{e1}$  处完全消失。镜头  $B(t)$  信号在  $t_{s2}$  处开始出现, 经过  $t_{e2}-t_{s2}$  长的线性增长后, 在  $t_{e2}$  处达 100%。生成的新镜头  $C(t)$  长度为  $t_{e2}+t_2$ , 渐变过程中的视频信号是镜头  $A(t)$  的衰减信号和镜头  $B(t)$  的渐增信号之和。若分别定义镜头  $A(t)$  和镜头  $B(t)$  的长度为  $t_{e2}+t_2$  的扩展镜头信号为:

$$A^*(t) = \begin{cases} A(t) & 0 \leq t < t_1 \\ A(t_1) \left( 1 - \frac{t-t_1}{t_{e1}-t_1} \right) & t_1 \leq t < t_{e1} \\ 0 & t_{e1} \leq t \leq t_{e2}+t_2 \end{cases} \quad (8-15)$$



$$B^*(t) = \begin{cases} 0 & 0 \leq t < t_{e2} \\ \frac{(t-t_{s2})B(0)}{t_{e2}-t_{s2}} & t_{s2} \leq t < t_{e2} \\ B(t-t_{e2}) & t_{e2} \leq t < t_{e2}+t_2 \end{cases} \quad (8-16)$$

则产生的新镜头  $C(t) = A^*(t) + B^*(t)$ 。

当  $A(t)$  或  $B(t)$  分别为零信号时，镜头渐变就退化为两种常见的视频编辑方式：淡入和淡出。

### 3) 镜头的检测：

(a) 切换镜头的检测。根据镜头切变的特点，切变检测显然应针对帧特性的突变来进行。为了降低计算量，加快检测速度，可以采用帧的直流分量 DC 图像来作为帧特性的描述。显然，在镜头内部 DC 图像变化很小，但在镜头切变处前后两幅 DC 图像分别对应于前一镜头的末帧和后一镜头的首帧，图像特性将会出现较大的变化。在 DC 图像的差序列中，这对应于一个单点峰值的出现。通过阈值检测，就可以确定切变的位置。但当图像中景物出现较大或比较明显的运动时，DC 图像的特性也会出现较大的变化，导致 DC 图像差序列中也出现一系列较大的值。这时，直接对 DC 图像差设定阈值将导致误检。但由于运动的连续性，这些较大值的出现也是连续的，而非单点峰值。针对这个特点，我们采用了对 DC 图像差计算如下定义的次大比值序列：

$$R2nd \max(t) = \frac{d_1(t)}{2nd \max(d_1(t'))} \quad (8-17)$$

[t-w, t+w]

并对此序列进行阈值检测的方法。其中  $d_1(t) = \|I_{DC}(t+1) - I_{DC}(t)\|$  为相邻帧的 DC 图像差序列， $w$  为窗口半径， $2nd \max(d_1(t'))$  为  $d_1(t)$  在以  $t$  为中心，以  $w$  为半径的窗口内的次大值。

公式 (8-18) 是针对理想镜头切变模型的，即帧内容的改变在一帧内完成。对实际视频信号镜头切变中可能出现的非立即转变情形，可以采用如下的比值序列

$$R4th \max(t) = \frac{d_1(t)}{4th \max(d_1(t'))} \quad (8-18)$$

[t-w, t+w]

计算当前 DC 图像差与窗口内第 4 大 DC 图像差的比值序列。对此序列进行阈值检测，可以得到镜头切变的开始位置和结束位置（相差 3 帧以内）。

(b) 渐变镜头的检测。在镜头渐变过程中各像素以恒定增量变化。针对此特点，可有如下的简便检测方法，首先定义间隔  $w$  帧的 DC 图像差序列  $dw(t) = \|I_{DC}(t+w) - I_{DC}(t)\|$ ，相对于渐变而言，在镜头内部 DC 图像  $I_{DC}(t)$  变化很小，可认为保持恒定，见图 8-9a)，这样，在序列  $d_w(t)$  上镜头渐变过程就对应于一个梯形形状的峰，如图 8-9b) 所示。其次，定义连续  $w$  帧的 DC 图像线性偏差序列

$$E_w(t) = \sum_{i=0}^{w-1} \left\| I_{DC}(t+i) - \left(1 - \frac{i}{w-1}\right) \cdot I_{DC}(t) - \frac{i}{w-1} \cdot I_{DC}(t+w-1) \right\| \quad (8-19)$$

在镜头内部，景物的局部变化使  $E_w(t)$  保持一个相对较高的水平。但在镜头转换处，信

号的变化以线性变化为主，将导致谷值的出现。

(2) 镜头组织。通过视频分割，连续的视频流被切分为可供检索的镜头。但对于许多视频信号来说，其所含的镜头数可多到上百上千，使得用户难于直接检索。以电影为例，有多于 500 镜头的影片并不少见，而现代动作片中镜头数则更多（如在“终结者 II”中，15min 的影片就有 300 个片段）。在电视节目的新闻报导中，也有大量的关于同一报导内容的镜头。在这种情况下，就需要对镜头进行组织，生成一种更高级、更具有语义性的结构（如场景 scene 或幕 act），以便于用户的检索。

场景转换图 STG (Scene Transition Graph) 是一种常用的镜头组织方法。该方法根据语义信息对镜头进行组织，从而对整个视频流进行建立一种层次化的表示。首先根据镜头间的相似性（主要利用颜色、形状及图像间相关等）对在内容描述空间对镜头进行聚类，再根据聚类结果利用时间轴上的约束生成场景转换图。

(3) 描述建立。与图像检索有所不同。视频检索的描述建立除了特征提取外，还包括镜头的代表帧生成。特征提取用于对分割得到的镜头建立描述，常用的镜头特征是镜头的主运动，这个特征适用于景物中有一个大背景的镜头。在此条件下不能得到满足（如景物中存在多个较大运动物体的情况）时，就需要对镜头根据运动特性对不同景物分层并对分层建立描述。代表帧生成用于对镜头生成一个或多个可代表内容的图像，可在浏览检索用来代表镜头的内容。

1) 主运动估计 (dominant motion estimation)。在多数视频图像中，往往有一个相对固定的背景，背景前面是运动的物体。此背景占据了大部分图像，其在图像平面中的变化是摄像机操作和运动（如聚焦、平移、追踪等）的结果。在这种情况下，主运动（即此背景的运动）估计无疑对图像的对准和拼接效果起着重要的作用。

不同算法所依据的运动模型不同。在背景景深变化不大的情况下，可以把整个背景作为二维平面来考虑，这样就得到了摄像机的二维运动模型。

对于背景景深变化较大的情况，此时单个二维运动模型已不能描述背景的运动，而必须考虑景深的影响。针对校准过的摄像机，提出了三维运动结构模型。

在模型参数的计算方面，为了去除前景运动的影响，在参数估计时一般不采用最小均方差 (least squares) 方法。而采用了 M 估计 (M-Estimation) 方法。

2) 层描述 (layered representation)。在许多情况下，景物由多个具有不同运动特性的较大物体组成，很难指定哪一个运动是主运动。在这种情况下，就需要采用层描述方法，根据运动特性的不同对景物进行分割。将其划分为不同的“层”，并分别对每层建立相应的描述。层描述一般包括三个部分：一个运动描述符的集合；对每个运动描述符在图像序列中与之对应即满足其约束关系的“层”；根据前两项及原始视频图像建立的对每个“层”建立的描述。

层描述所要解决的问题是：需要多少个运动模型来描述景物中的运动；运动模型参数的计算；每个运动模型支撑“层”的计算。为了对运动模型的数目加以限制，算法一般都采用 MDL (Minimum Discription Length) 准则来进行计算。

在 QBIC 的实现中，就是根据 2D 运动模型，通过不同物体（由于在景物中远近不同所引起）的仿射变换的参数不同，将景物划分为不同的层 (layer)，每一层具有自己的 2D 仿射运动参数以及在每一帧中的对应区域，这样，就对整个片段建立了一个运动的分层表述

(layerd representation), 供检索时使用。

3) 镜头代表帧生成。在许多视频检索应用中, 浏览检索是一种十分重要的检索手段。在这种检索方式中, 镜头的内容用一幅或多幅能够代表镜头内容的图像(称为代表帧 representative frame, 简记为 r-frame)来表示。系统通过在检索界面上提供各镜头的代表帧, 使得用户可通过浏览来选择相应的镜头。这样, 如何生成代表帧, 使其更好地表现镜头的内容, 就成为一个需要解决的问题。

最简单的办法是直接从镜头的帧序列选择一幅或多幅图像作为代表帧。常用的方法是选择图像的首帧、中间帧和/或末帧, 但这种方法的缺点是缺乏灵活性, 不能根据镜头的内容变化选择代表帧, 还可以采用根据对帧相关差(对应像素差)设阈值来选择代表帧的方法。

另一种常用的代表帧生成方法是图像拼接(mosaicing), 即对分割得到的镜头融合镜头中多帧的背景信息生成一个全景的拼接图像(mosaic)来代表镜头内容。

#### 4. 基于内容检索索引方法

在基于内容检索系统中, 快速索引技术是影响系统速度的关键。由于媒体对象的特征描述一般都是高维的, 而特征空间中的相近意味着较大的相似度; 因此问题的关键就在于如何在高维空间点集中寻找与给定点距离最近的一组点。在计算机科学中, 在点集中寻找与给定点距离最近点的问题叫做最近邻搜索(nearest-neighbor-search)问题, 记作 NN-Search 问题。基于内容检索的索引问题要求寻找与给定点距离最近的一组点(不妨设为  $k$  个), 常记作  $k$ -NN-Search 问题。

在一维空间下, 最近邻搜索可采用对顺序表进行简单的折半查找来完成, 但在高维空间这种算法就难以直接应用了。最早的最近邻查找算法是所谓的桶算法(bucketing algorithm)。这种算法将数据集空间进行规则划分(如成格状), 每个划分中的数据点存于一个桶中(bucket)。这样, 最近邻搜索就转化为在所有的桶中寻找与给定点最近的桶。由于桶算法对数据集空间进行规则划分, 因此其对于均匀分布的数据集十分有效。

对于主存中的最近邻搜索来说, 一种比较实用的索引方法是  $k$ -d-树方法, 其思想是按照一定的准则选择某一坐标轴方向作为切分方向, 将数据集切分为两个子数据集, 再对此两个子数据集递归切分, 形成一棵检索树。优先  $k$ -d 树(priority  $k$ -d tree)这种索引结构不仅检索速度快, 适用于  $k$ -NN-Search 问题, 而且其空间复杂度与数据集的维数成线性关系, 且与二级存储器实现相容, 是目前解决  $k$ -NN-Search 问题最有效的一种索引算法。

近似最近邻搜索(approximate nearest neighbor search)也是基于内容检索索引研究的一个热点。由于在多数应用中, 用户只要求检索结果中有一个或几个满意媒体对象即可, 而并不要求查询结果一定是信息库中满足相似度和检索结果集合大小限制的所有对象。相反地, 在多数情况下用户希望牺牲检索的精确度以换取更高的检索速度, 这就导致了近似最近邻搜索技术的产生。而大量的实验表明, 对于高维数据集来说, 即使允许极小的近似(检索误差), 也会对检索速度带来很大的提高。

在基于内容检索系统中, 为加快检索速度而经常使用的另一个技术是分层索引技术。对于许多检索操作(如二次型距离的直方图匹配)而言, 特征间距离的计算往往需要很大的计算量。在这种情况下, 可采用一种计算量较小(cheap)的距离度量先对信息库进行检索, 得到一组候选的媒体对象集, 再对此候选媒体对象集采用原距离度量, 以此来降低计算量。

8.2.3 一个基于内容检索系统的设计和实现

随着多媒体技术的迅速普及，Web 上将大量出现多媒体信息，例如，在遥感、医疗、安全、商业等部门中每天都不断产生大量的图像信息。这些信息的有效组织管理和检索中都依赖基于图像内容的检索。目前，这方面的研究已引起了广泛的重视。例如，由 IBM 公司开发的 QBIC 是最有代表性的系统，它通过友好的图形界面为用户提供了颜色、纹理、草图、形状等多种检索方法；美国加州大学伯克利分校与加州水资源部合作进行了 Chabot 计划，以便对水资源部的大量图像提供基于内容的有效检索手段。此外还有麻省理工学院的 Photobook，可以利用 Face, Shape, Texture, Photobook 分别对人脸图像、工具和纹理进行基于内容的检索，在 Virage 系统中又进一步发展了将多种检索特征相融合的手段。澳洲的 New South Wales 大学已开发了 NUTTAB 系统，用于食品成分数据库的检索。

为了阐述基于内容检索系统的工作原理，我们举下述一例：清华大学研制的 Internet 上静态图像的基于内容检索的原型系统。

1. Internet 上静态图像的基于内容检索的原型系统

清华大学计算机科学与技术系结合国家 863 高技术研究发展项目“Web 上基于内容的图像检索”的研究，于 1997 年研制了一个 Internet 上的静态图像的基于内容检索的原型系统。该项目的研究目标是开发能在 Internet/intranet 环境下，通过友好的人机界面，以颜色、纹理等图像特征或样本图像检索图像的方法和工具。目前该系统对 12000 张景物照片图像的高维特征建立了 GSS-索引，采用 Java 语言来实现人机交互模块，通过浏览器为检索用户提供一个友好的人机交互界面，得到的检索请求转换为 CbExpr 检索表达提交给检索引擎进行处理。与网络所造成的延时相比，检索时间基本上可以忽略，表现出了相当好的实时性。

(1) 系统结构。设计系统结构是根据 Web 技术和信息检索技术的实现特点，划分为检索相关模块和建库相关模块两部分，结构框图如图 8-9 所示。

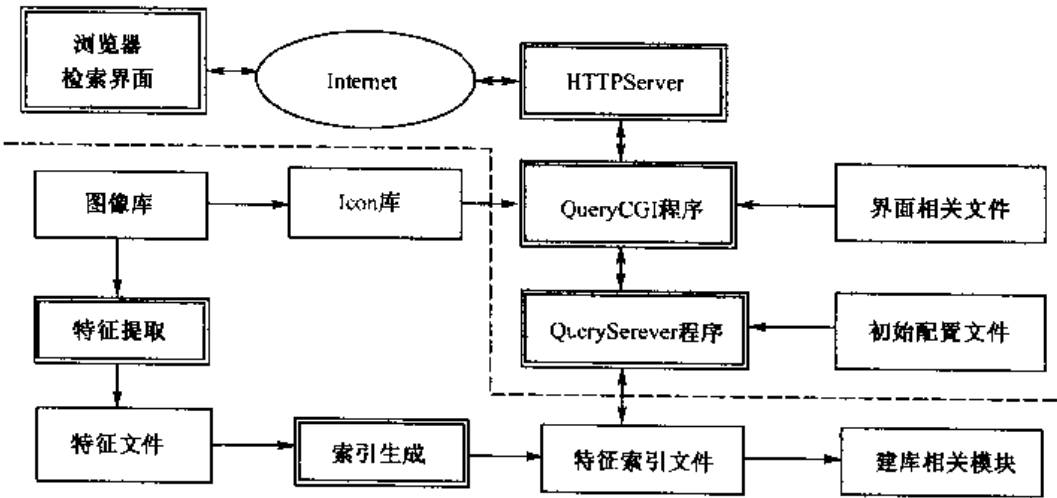


图 8-9 Web 上基于内容的图像检索原型系统的系统结构

1) 检索相关模块。这个模块包括用户检索时用到的程序和文件，包括 Querycgi 程序、

Queryserver 程序、初始配置文件和一些界面相关文件。

在检索时，用户通过本地的浏览器，通过 Http Server 访问 Querycgi，并提出检索请求。Querycgi 为一个 CGI 程序，通过 CGI 接口与 Http Server 通信。它负责提供检索界面，接受用户的检索请求，将检索请求传送给 Queryserver 程序并将 Queryserver 程序返回的检索结果以可视的形式（图像的 Icon 或图像本身）传给用户。Queryserver 程序为一个“精灵”（daemon）程序，绑扎于指定端口上，接受 Querycgi 传来的检索请求，检索相应的索引文件，处理后返回检索结果。

界面相关文件是一些与检索界面有关的文件（如构成检索界面的 Java Class 文件，利用模板提供特征时的模板特征文件等）；初始配置文件则是提供 Queryserver 参数（如绑扎端口，各特征对应的索引文件名）的文件，供 Queryserver 启动时初始化使用。

2) 建库相关模块。建库相关模块主要是用来建立和维护整个图像库和相关文件的，主要包括特征提取和索引生成两部分。

特征提取模块用来对图像库中的图像完成特定特征的抽取，并生成相应的特征文件；而索引生成模块则用来对得到的特征文件进行处理，根据相似检索（similarity indexing）原理生成相应的特征索引树和特征索引文件，供检索时使用。此外，建库相关模块还包括一些维护图像库（如完成图像格式转换，生成 Icon）的工具。

(2) 检索查询界面设计。友好的人机交互界面是一个成功检索系统不可缺少的条件，它可以大大提高检索的效率。在基于内容检索中，由于特征值为高维向量，不具有直观性，因此必须为其提供一个可视化的输入手段。目前采用的特征包括主颜色、颜色直方图、纹理分布、草图等 7 种，根据特征的种类不同，采用了两种特征输入手段：

1) 操纵交互输入方式。适用于具有较强操纵性的特征。在这种方式中，用户可通过操纵工具来调整特征值，而特征值的变化直接向用户体现出来。通过这种方式，用户可得到想要的特征值，并提交来构成检索表达。如通过滑动条来调节主颜色，通过画笔来给出颜色直方图构成和颜色分布等。

2) 模板选择输入方式。有一些特征难以对其进行操纵和调节（如纹理），因此也难以通过操纵交互输入方式来进行特征输入。针对这类特征，我们提供一组在特征空间中具有代表性的特征模板，用户可通过选择与检索特征最接近的模板来完成特征值的输入。

同时还在研究实现下述特征输入手段：

1) 浏览检索。系统提供一组对象，用户从中选择与检索目标最接近的对象。系统根据用户的选择再给出一组对象供用户进一步选择。如此往复，直到检索完毕。

2) 样本检索。即由用户提交一幅图像，系统检索图像库中与此图像相似的一组图像。

系统的界面模块分别实现了七种特征的前端处理界面，即主颜色、纹理、色调直方图、轮廓、颜色分布、彩色直方图和主题。界面设计如图 8-10 所示。

界面结构分为三部分组成：

1) 检索特征预览。全部提供了七种特征，在任意时候只显示其中三个（如图 8-10 所示），用户可用鼠标拖动下面的滑动条以显示别的特征的当前查询值。对每一种特征，都有相应的特征名、使能选中框和权重滚动条（滚动条前面的编辑框显示当前的权重值，也可以直接键入）。用户必须选中该特征的使能框，权重滑动条和对应的特征输入区才可用。在该特征使能时，鼠标单击这一区域即可激活对应的特征输入模块。

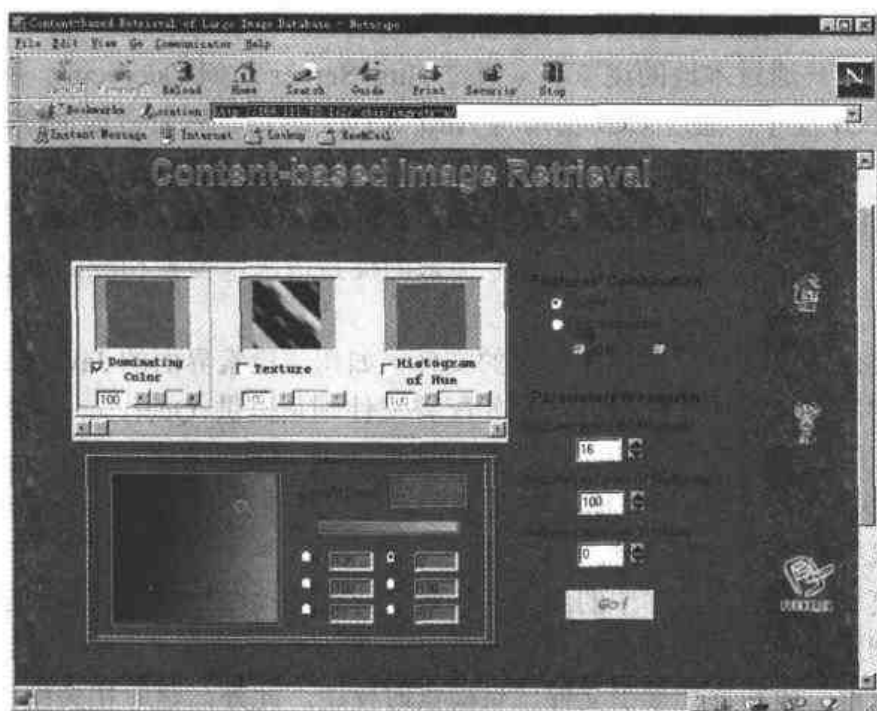


图 8-10 基于内容检索系统的主界面

2) 检索控制参数。用户可以选择三种单特征查询组合方式：与、或、权重调节中的一种。它们直接影响对单特征查询结构集的处理：

- 最大返回结果数。由于图像检索是近似检索，所以查询结果一般都不止一个，我们将满足条件的图片按其相似度从大到小排序，返回前面的若干个。我们规定其合法值是 16 到 100。
- 最小相似度。图片库中的任意图片与查询条件的相似度在区间 $[0, 1]$ 上，所以应该预定一个阈值，相似度大于它的图片才认为满足条件。在界面上，用户可以输入 0 到 100 之间的一个整数。
- 检索精度 (epsilon) 合法值是 0 到 100。在后台处理时，将它映射到 $[0, 1]$ 区间内。

3) 特征输入区 (工作区)。基于内容的图像检索的查询条件提交方式不同于传统的文本检索，针对每一种特征都应设计特定的交互方式。

(3) 提交特征分析：

1) 主颜色 (Dominating Color)。主颜色就是图片中占较大比重的颜色。在图像处理中，主颜色的提取过程是：将图像的各点颜色值 (一般是 RGB 值) 量化到  $16 \times 16 \times 16$  的区间内，然后对图像的点聚类。如果某聚类颜色的像素点占图像全部像素的 20% 以上，就认为该聚类中心颜色是该图的主颜色。所以，一幅图片的主颜色可能不止一种。

2) 纹理 (Texture)。纹理特征适用于墙纸、布匹等具有简单、重复性的图案的检索。它的提取是用 Gabor 算子分别从 5 个方向计算得到 6 个值，总共 30 个坐标。对纹理而言，如果用户没有一定的美术基础，很难直接用绘画方式提交数据，因此采用了预定义模板的方式，用户可以在特征输入区中浏览选择。目前提供了六个模板，以后还可以随时增减，这六个模板的选择是根据图像库中的常见纹理种类面定的。

3) 色调直方图 (Histogram of Hue)。在常用的彩色空间中，除了 R、G、B 还有 H (Hue

色调)、S (Saturation 饱和度)、B (Bright 亮度)。其中 Hue (色调) 的合法取值是 0 到 360, 从视觉效果来讲, 色调的变化范围就是: 赤、橙、黄、绿、青、蓝、紫七色的逐渐过渡。图片的色调直方图表示了各种色调在其中的比例关系, 这种关系有较大的变化范围, 不宜使用模板, 同时要求普通用户也能容易地表达查询要求, 因此实现的是一个类似于 Paint (画板) 的特征提交界面。用户只需很简单的操作就可以画出若干彩色块, 大致地从它们的面积就可以知道色调的分布。在正式提交查询数据时, 前端模块还对整个画板的像素进行量化、统计, 向服务器提交的是一个 32 维的直方图向量。

4) 彩色直方图 (Histogram of HSB)。H、S、B 直方图特征的“工作区”与色调直方图的完全一样, 只是后台处理不一样。在提交最后的数据时, 前端模块先将各像素值从 RGB 彩色空间转化到 HSB 彩色空间, 然后量化到  $4 \times 4 \times 4$  的范围内, 同时进行统计, 最后向服务器提交的是一个 64 维的直方图向量。由于 HSB 的比例关系与人的直观感觉不太一致, 所以这种特征的查询效果并不是很好。

5) 颜色分布 (Color Distribution)。颜色分布是另外一个检索效果较好的特征。它同时表示了图像中颜色的值与位置属性, 比较直观, 在查询界面上以用户直接画图的方式提交数据。其工作区的外观和使用与色调直方图一样, 但颜色分布特征的提取是这样的: 将用户所绘制的图像分割成  $6 \times 6$  个大小一样的“Icon”, 对每一个 Icon 中像素的 R、G、B 值分别求平均, 这样得到了 36 个“平均像素值”, 其中既包含了位置的信息, 又有颜色值的信息。最后, 前端模块向服务器提交的是一个  $6 \times 6 \times 3 = 108$  维的向量。例如, 用户想查询有“蓝天与大地”内容的图片, 他可以简单地画出图 8-11 的一幅画, 得到的结果如图 8-12 所示。可以看出, 颜色分布特征的查询效果是比较好的。



6) 轮廓 (Sketch)。在图像库中有很多图片的内容难以用颜色、位置、纹理来描述。比如想查询“画面分为上下两部分”的图像,用前面提到的特征无法方便地描述,可以用 Sketch 特征来描述画面中的各个部分的相对位置关系。提取特征的步骤是:先用颜色聚类的方法对画面进行划分,然后用中值滤波去掉噪声,得到若干均匀的颜色块(称为“Segment”)。接着,提取图像的边缘特征,得到一个二值图像,最后将它缩小为  $16 \times 16$  的大小,取得一个 256 维的特征向量,其中每一维的值或为 1 或为 0。对该特征的提交,我们仍采用了用户直接绘画的方式,因为它的变化多,语言表示不直观。“工作区”类似于色调直方图,但是绘制的是黑白图像,并且基本构图元素是点、直线、矩形、椭圆、多边形等线状图形,而不是块状图形。数据的预处理方法是:将用户所画的图形分割成  $16 \times 16$  个区域,对区域内的像素值进行统计,如果黑色像素的个数大于阈值,则认为该区域为黑色,否则认为是白色。这样,我们可以得到一个 256 维的查询特征值。作为例子,假设现在提交一幅图(图 8-13),得到的结果如图 8-14,总的效果还是比较好的。

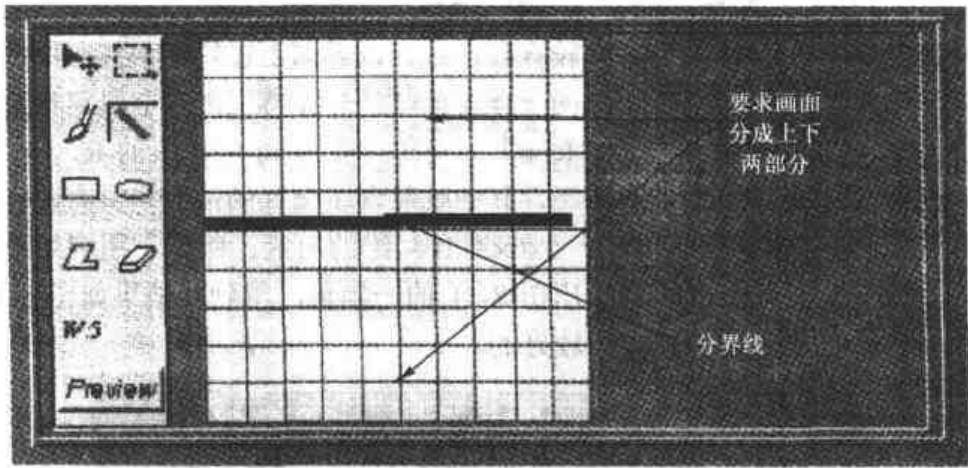


图 8-13 提交的轮廓特征



图 8-14 轮廓特征的查询结果

7) 主题 (Topic)。在研究中我们注意到,基于内容的图像检索也有它的局限性,其中一个很大的不足是难以表达抽象概念,而这一点正是基于文本的检索的长处。比如说查找有“兔子”的图片,用目前我们实现的特征无法表达这一概念,而用文本检索则只需两个字。所以,我们将基于内容和基于文本两种检索方法结合起来,在前六种图像特征之外增加了“主题”检索。我们把主题特征视为图像特征之一,除了索引文件和查询方法以外,在对结果集



的处理方面（如与、或、权重调和）与其他特征完全一样，这样既增强了系统的查询能力，又保证了结构的模块化。值得注意的是，主题特征也有“相似度”的定义。因为很难把握两个词之间的相似程度，我们目前只定义了一种比较简单的“相似度”：

如果用户提交的关键词个数为  $M$  个，图像  $i$  的主题描述中有其中的  $N$  个关键词，则图像对查询条件的相似度  $SF(I) = N/M$ 。

可以事先对每一幅图片都用几个关键词来加以描述，然后对所有图片的关键词进行归类和合并，构成一个有着树型结构的关键词集。由于关键词集是确定的，所以在向服务器提交查询数据时只提交关键词的 ID 号即可。用户可以在其中选择若干个关键词，综合描述一幅图像。

主题词和图像特征查询的结合能取长补短，充分发挥前者表达抽象概念、后者表达形象概念的优点，获得更好的查询效果。

#### （4）检索关键技术：

1) 检索表达。检索表达能力是影响一个基于内容检索系统性能的重要因素。目前实现的基于内容检索系统，一般都通过指定特征值来检索内容描述与此特征值最近的一组对象，表达能力有限，不能表达在内容描述空间中具有复杂相似度函数分布的检索概念（如“日落”概念在颜色直方图空间对应的在红、黄或紫颜色区域呈峰值）。为此，项目组提出了一种可表达内容描述空间中复杂概念的基于内容检索表达机制 CbExpr，并实现了基于 CbExpr 检索表达机制的图像检索系统。实验表明，CbExpr 具有强大而灵活的表达能力，取得了较好的效果。

2) 快速索引算法。快速索引结构和快速索引算法是基于内容检索的内核，也是大容量基于内容检索系统实用化的关键。其基本思想源于计算机科学中的最近邻搜索（nearest neighbor search）问题，并在地理信息系统（GIS）等信息系统中得到了广泛的应用。目前，常用的快速索引结构主要包括以 R+树为基础的索引树和以 k-d 树为基础的索引树。

项目组提出并实现了一种通用快速索引结构 GSS，通过对此通用快速索引结构的分割方法、闭包结构和距离定义进行不同的选择，可分别实现传统最近邻索引算法中的以 R+树为基础的索引树和以 k-d 树为基础的索引树，还可以进行采用二次型距离定义的检索处理。而通过采用复合闭包，可对数据集给出更多约束，从而更有效地降低检索过程中的搜索量，提高检索速度。

针对图像库基于内容检索系统的应用背景，我们实现了基于此通用快速索引结构的索引生成模块、索引存盘模块、索引读取模块和索引检索模块。其中，索引生成模块用于从特征抽取得到的特征列表文件生成检索树；索引存盘模块用于将此索引树存成磁盘上的索引文件；索引读取模块的作用则相反，用来从索引文件恢复索引树；索引检索模块用于处理对于索引树的检索，给定参数（检索特征值、目标个数、最小相似度、近似度）后返回一组对象的标识及其对应的相似度值。

## 8.3 小结

本章讨论了多媒体数据库及基于内容检索，对有关多媒体数据的存储问题、管理问题和多媒体数据库的体系结构、多媒体数据库基于内容检索的关键技术和基于内容检索系统的设计原理和实现技术等均作了分析和讨论。

本章是多媒体技术在有关领域的应用技术，通过本章的学习可以进一步了解到多媒体技术的应用是非常广泛的，而在实际应用中涉及了有关学科的知识和技术。

## 8.4 习题

1. 基于内容检索的体系结构可分为哪两个子系统？  
(A) 特征抽取和查询子系统。  
(B) 多媒体数据管理和调度子系统。  
(C) 用户访问和数据库管理子系统。  
(D) 多媒体数据查询和用户访问子系统。
2. 基于内容检索要解决的关键技术是\_\_\_\_\_。  
(1) 多媒体特征提取和匹配。  
(2) 相似检索技术。  
(3) 多媒体数据管理技术。  
(4) 多媒体数据查询技术。  
(A) (3) (4) (B) (2) (3)  
(C) (1) (2) (D) 全部
3. 简述多媒体数据库中基于内容检索系统的工作原理。

## 8.5 参考答案

1. (A)
2. (C)

3. 答：多媒体数据库基于内容检索系统的工作原理概述如下：

基于内容的检索作为一种信息检索技术，接入或嵌入到其他多媒体系统中，提供基于多媒体数据库的检索体系结构如图 8-5（基于内容检索的体系结构）所示。

由图 8-5 可见，基于内容检索系统分为两个子系统：特征抽取子系统和查询子系统。系统包括如下功能模块：

(1) 目标识别。为用户提供自动半自动识别静态图像、视频、镜头的代表帧，是用户感兴趣的内容或区域。

视频序列图像动态目标，对目标进行特征抽取、查询，处理进行整体的或局部的内容检索，可采用全局特征或局部的特征。

(2) 特征抽取。提取用户感兴趣的又适合于基于内容检索的特征。如颜色分布情况、颜色的组成情况、纹理结构、方向对称关系、轮廓形状大小。

(3) 数据库。多媒体数据库，声、文、图；特征库，预处理特征；知识库；知识表达。

(4) 查询接口。有三种输入方式：交互输入方式；模板选择输入方式；用户提交特征样板输入方式。多媒体特征组合功能和查询结果浏览。

(5) 检索引擎。利用特征之间的距离函数来进行相似性检索。对于不同的特征用不同的相似性测度算法，检索引擎中系统有效的是相似性测度函数集。

(6) 索引/过滤。通过索引和过滤达到快速搜索的目的。把全部的数据通过过滤器变成新的集合再用高维特征匹配来检索。

基于内容检索的工作过程包括以下几个步骤：

- 1) 提交查询要求：利用系统人机交互界面输入方式形成一个查询条件。
- 2) 相似性匹配：将查询特征与数据库中的特征按一定的匹配算法进行匹配。
- 3) 返回候选结果：满足一定相似性的一组候选结果按相似度大小排列返回给用户。
- 4) 特征调整：对系统返回的一组初始特征的查询结果，用户通过浏览选择满意的结果，或进行特征调整，形成新的查询，直到查询结果满意为止。

基于内容检索的工作过程如图 8-6 所示。

## 第9章 多媒体会议系统的开发和交互式电视技术

### 本章要点

- 多媒体会议系统的开发主要涉及到：视频会议系统的结构及标准、综合业务多媒体通信终端的设计和实现、多点控制单元、视频会议系统的服务质量（QoS）及资源管理和安全保密问题。
- 交互式电视技术是近几年的热点课题是交互式电视技术 ITV（Interactive Television），这是因为交互式电视技术有较好的经济效益、社会效益和广阔的应用前景。

### 9.1 多媒体会议系统的开发

当今信息社会的发展对通信提出了更高的要求，人们已不满足简单的语音和文字通信，希望集语音、文字和图像于一体的多媒体通信。这是继电报、电话、传真及电子邮件（E-mail）后又一新的通信手段。视频会议系统（Video Conference）就是这种新型通信手段之一，它可以点对点通信，也可以多点对多点的通信，它在同一传输线路上承载了多种媒体信息：视频、音频和数据等，实现多点实时交互式通信，同时也可以将不同地点与会人员的活动情况、会议内容及各种文件以可视新闻的形式展现在各个分会场，这是一种快速高效、日益增长、广泛应用的新的通信业务。

#### 9.1.1 视频会议系统的结构及标准

根据通信节点的数量，视频会议系统可分为点对点视频会议系统和多点视频会议系统。

点对点视频会议系统支持两个通信节点间视频会议通信功能，它的主要业务是：

##### 1. 可视电话

可视电话是在现有公共电话网上使用的具有双工视频传送功能的电话设备。由于电话网带宽的限制，可视电话只能使用较小的屏幕和较低的视频帧率。例如使用 3.3in 的液晶屏幕，每秒钟可传送 2~10 帧画面。

##### 2. 桌面视频会议系统

这种视频会议系统利用用户现有的台式机（如 PC 机）平台以及网络通信设备和远地另一台装备了同样或兼容设备的台式机通过网络进行通信，这种系统仅限于两个用户或两个小组用户使用。Intel 公司的 Proshare Personal Conferencing Video System 200 是这类系统的一个典型示例，这是一种点对点的个人视频会议系统，支持 ISDN 和 LAN 的连接，采用硬件编码压缩，软件解压缩，为了方便协同工作，Proshare 还提供共享笔记本和共享应用程序。

##### 3. 会议室型视频会议系统

在会议室型视频会议系统的支持下，一群与会者集中在一间特殊装备的会议室中，这种

会议室作为视频会议的一个收发中心，与远地的另外一套类似的会议室进行交互通信，完成两点间的视频会议功能。由于会议室与会者较多，因此对视听效果要求较高，一套典型的系统一般应包括：一台或两台大屏幕监视器、高质量摄像机、高分辨率的专用图形摄像机、复杂的音响设备、控制设备及其他可选设备，以满足不同用户的要求。

多点视频会议系统允许三个或三个以上不同地点的参加者同时参与会议。多点视频会议系统一个关键技术是多点控制问题，多点控制单元 MCU (Mulitpoint Control Unit) 在通信网络上控制各个点的视频、音频、通用数据和控制信号的流向，使与会者可以接收到相应的视频、音频等信息，维持会议正常进行。

(1) 视频会议系统的结构。视频会议系统的结构如图 9-1 所示，它主要由视频会议终端、多点控制器、信道(网络)及控制管理软件组成。

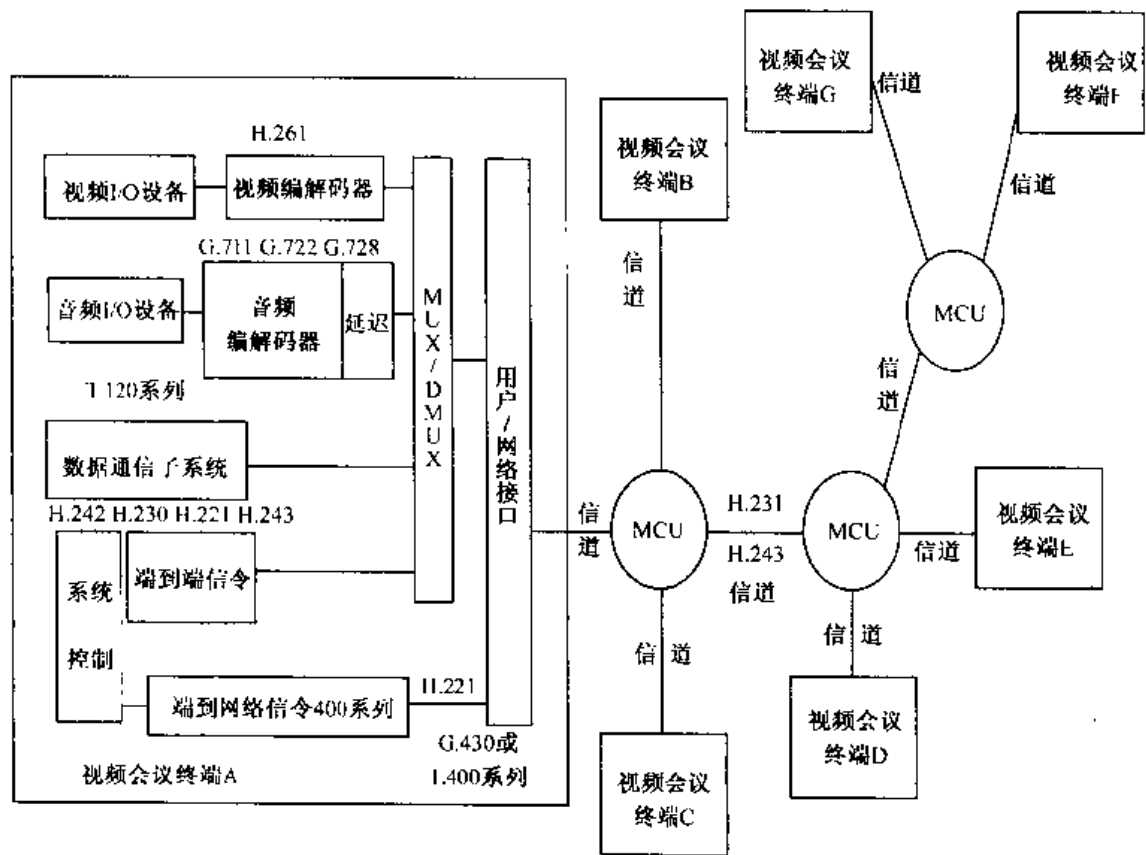


图 9-1 视频会议系统结构框图

视频会议系统终端的主要功能是：完成视频信号的采集、编辑处理及显示输出、音频信号的采集、编辑处理及输出、视频音频数字信号的压缩编码和解码，最后将符合国际标准的压缩码流经线路接口送到信道，或从信道上将标准压缩码流经线路接口送到终端中。此外，终端还要形成通信的各种控制信息：同步控制和指示信号、远端摄像机的控制协议、定义帧结构、呼叫规程及多个终端的呼叫规程、加密标准、传送密钥及密钥的管理标准等。

多点控制单元 MCU 是视频会议系统的关键设备，它的主要功能是对视频、语音及数据信号进行切换，例如它会把传送到 MCU 某会场发言者的图像信号切换到所有会场。对于语音信号，若同时有几个发言，可以对它们进行混合处理，选出最高的音频信号，切换到其他

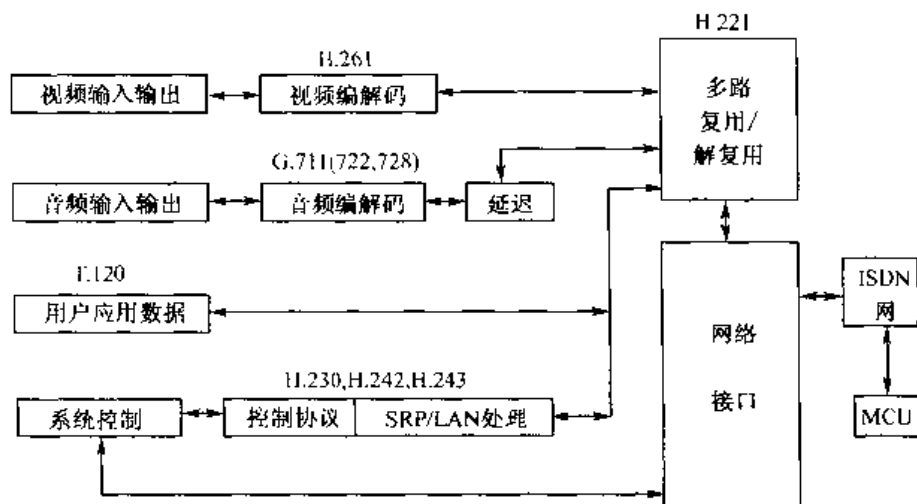


图 9-2 H.320 标准系统框图

H.230 传递帧同步控制和指示信号，负责处理基于 H.320 的 CODEC 设备之间传送的控制信息。

H.242 描述了在高至 2M 的数字信道上，会议电视终端之间建立通信和设置呼叫的规程，定义了基于 H.320 设备之间传送压缩视频和音频信号的协议。

H.261 又称为 Px64，它是视频编解码器的标准，采用中间格式兼容不同电视制式间的差异，是一种有运动补偿的帧间预测编码+变换编码（ZDDCT）+量化+可变长编码+传输缓存器控制的混合编码方式。视频编码器按照图像内容进行帧内/帧间判决和处理。

G.711 64Kbit/s PCM 电话质量（3.5kHz）语音压缩标准。

G.722 48/56/64Kbit/s ADPCM 高保真质量语音压缩标准。

G.728 16Kbit/s LD-CELP 语音压缩标准。

H.231 定义了点控制单元及如何连接 3 个或更多的基于 H.320 CODEC 设备。H.243 主要处理多个终端之间建立通信的过程，它定义了 H.320 CODEC 与 MCU 之间控制过程。H.231 和 H.243 两个建议主要针对视听业务，对数据仅提供初步的支持。这两个建议的控制功能比较简单，有些专家认为这是多点通信控制的暂时解决方案。按照 H.231、H.243，要实现对多个视听业务终端的多点控制，需通过设计 MCU 桥接各个视听业务终端，MCU 将向各终端发送 SCM（选定通信模式）信号，并能实现对图像画面，声音等选择、切换功能，MCU 中有分接器与按控制要求的复接器。

H.244 是使用 H.221 LSD/HSD/MCP 信道的远端摄像机控制协议。

H.281 与 H.224 相同也是远端摄像机控制协议，只是采用数据链路协议。

H.233 提供了 H.320 设备加密标准。

H.234 确定了如何在不同点之间传送密钥的密钥管理标准。

2) H.323 系列标准。H.323 描述无 QoS 保证的 LAN 多媒体通信终端、设备和服务。H.323 终端、设备可携带实时语音、数据、视频或它们的任意组合。其中无 QoS 保证的 LAN 包括 Ethernet, Fast Ethernet, FDDI, Token Ring 等。H.323 的 LAN 可以是一个网段或环，也可以是复杂拓扑结构的多个网段。但多个网段（包括 Internet）可能会导致很差的性能。H.323 终端可以是计算机或单独的设备（如视频电话），其中语音是必须支持的，数据和视频则是

可选的。该标准系统框图如图 9-3 所示。

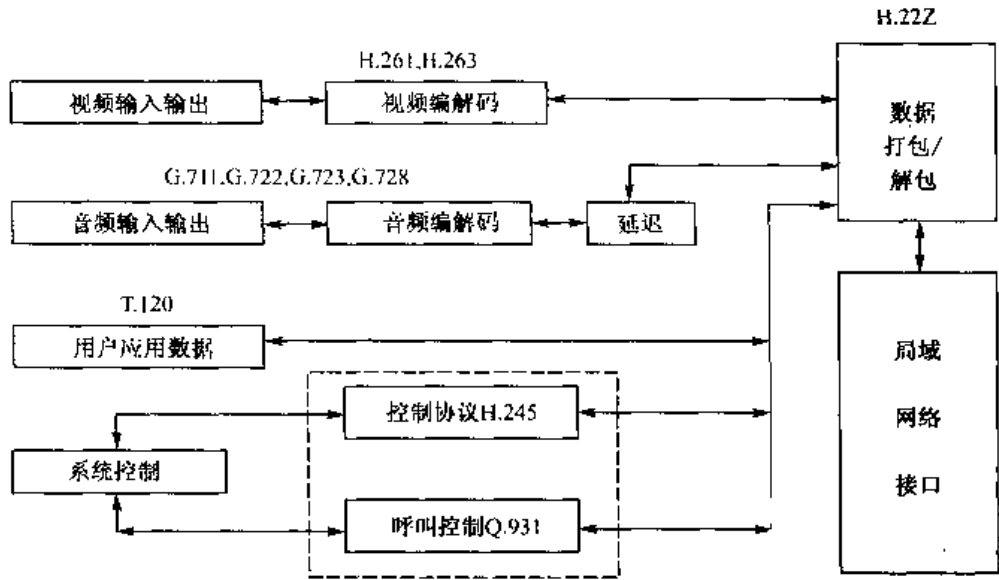


图 9-3 H.323 标准系统框图

H.323 系列包括：

● 视频：

H.261 P×64Kbit/s 下音频视频业务的视频编解码器。

H.263 低速率通信的视频编解码器。

● 音频：

G.711、G.722、G.728 同前。

G.723 5.3Kbit/s 和 6.3Kbit/s 多媒体通信的双速率语音编码，将改名 G.723.1。

● 复用：

H.222 无 QoS 保证 LAN 下媒体流的打包和同步。

● 控制：

H.230 视听系统中央同步控制和指示信号。

H.245 多媒体通信控制协议。

H.323 的网络接口要求能提供 H.225.0，H.222 定义的服务的网络接口，如 TCP，SPX，UDP，IPX 等。

H.323 与 ITU 定义的各会议系统之间的互操作性如图 9-4 所示。

其中：

H.310：H.231—ISDN。

H.324：GSTN（General Switch Telephone Network）普通交换电话网。

H.320：ISDN

H.322：QoS 保证的 LAN 上多媒体终端和系统。

H.323：无 QoS 保证的 LAN 上多媒体终端和系统。

GW：网关单元（Gate Way unit），提供 LAN 上 H.323 终端与其他 ITU 终端或其他 H.323 网关的通信。

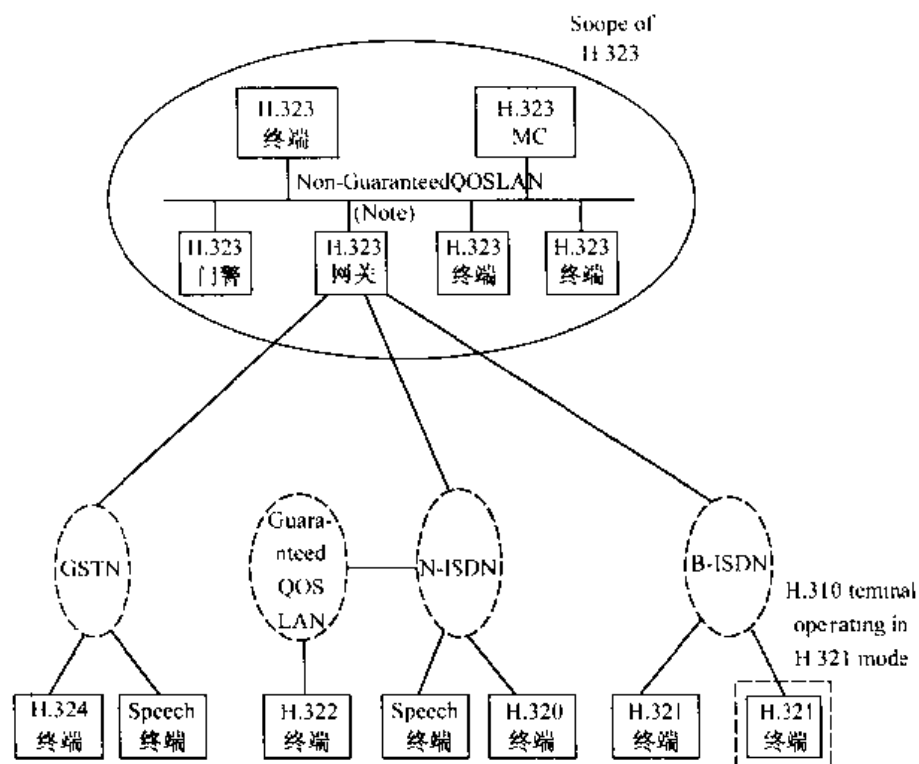


图 9-4 各 ITU 标准间的互操作

3) H.324 系列低速率通信标准。H.320 标准虽然在今天的电视会议系统中被广泛采用，但因为其只适合于 ISDN，E-1，T-1 等高速率的数字网，而并不适合于像电话线这种窄带网上使用，因此 ITU-T 又着手制定了适合于极低速率通信网络的 H.324 标准。其系统框图如图 9-5。

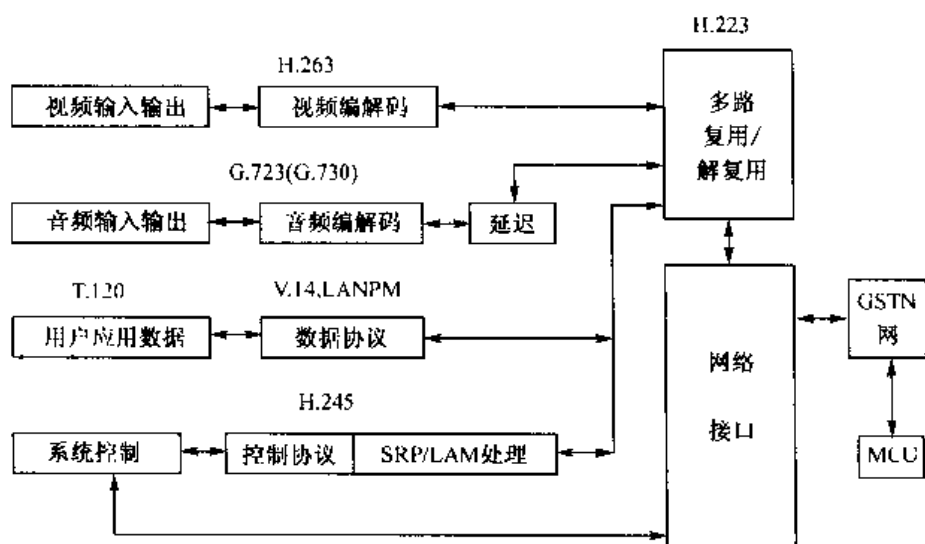


图 9-5 H.324 标准系统框图

H.324 低速率多媒体通信终端 (Terminal for Low Bitrate Multimedia Communication) 用于在 GSTN (General Switch Telephone Network) 上用 V.34 Modem 传输实时语音、视频、数据的会议系统标准。



H.324 标准集包括如下标准:

- 视频。H.263 低速率通信的视频编解码器。
- 音频。G.723 5.3Kbit/s 和 6.3Kbit/s 多媒体通信的双速率语音编码, 将改名 G.723.1。
- 复用。H.223 低速率多媒体通信复用协议。
- 控制。H.245 多媒体通信控制协议。
- 安全保密: H.233 视听服务的安全系统; H.234 视听服务的密钥管理和验证系统。

表 9-1 描述了系列国际标准的组成部分及应用场合。

表 9-1 系列国际标准的组成及应用

终端类型	H.320 系列	H.324 系列	H.323/ H.322 系列	H.321 系列	机 顶 盒
支撑环境	N-ISDN	PSTN/PSDN	LAN	ATM B-ISDN	HFC, PTTC
音频编码	G.711 G.722 G.728	G.723 AV.25Y	G.711, G.722 G.728	G.711, G.722 G.728 H.262	H.262
视频编码	H.261	H.263 H.264	H.261	H.261 H.262	
复用	H.221	H.223		H.221 H.222.0 H.222.1	H.222.0 H.222.1
网络接口	N-ISDN	V.34	LAN	AAL ATM	电缆 MODEM

4) T 系列标准。T.120 是国际电信联盟通信标准部开发的系列国际标准, 此标准是为多点和多媒体会议系统中发送数据而制定的。T.120 也为连接白板和非会议电视应用及文件传输提供了应用规范。

T.120 模型规定了一个通信的基础结构和在此基础上的应用规程。该模型遵循 ISO 开放系统互连七层模型, 每一层向上一层提供业务并向上一层发送由低层提供业务使用的数据。T.120 通信基础结构设计灵活, 它能同时处理多个独立的应用, 允许与电路交换的通信网络和基于分组的局域网以及数据网任意组合进行连接。

T.120 由三部分组成:

(a) T.123 传输规程框架规定了不同的网络之间的连接, 包括在物理层使用的 V 系统标准调制解调器的公共交换网; 使用 X.25 规程的分组交换数据网; ISDN 和电路交换数据网, 网上的语音、电视和 T.120 数据根据 H.221、H.320 标准内的成帧规程复用。LAN 规程不在 ITU-T 权限内, 但对于 TCP/IP 和 NOVELL 公司的 IPX 的参考框架已开发出来, 它包括一个增强的传输层能力, 因此起到了面向连接的网络作用。

(b) 多点通信业务 (MCS), 包括 T.122 业务定义和 T.125 规程规范。MCS 提供面向连接的业务, 它与 T.123 传输堆栈操作无关。MCS 在 OSI 模型的会话层, 将多个点对点传输连接分到多点通信领域。在 MCS 域内的节点按层次组织, 向上连到多点控制单元 (MCU), MCS 可以是台式的或独立设备中的一个独立节点, 如会议桥。

数据要沿最有效的路径送到节点, 多点通信业务 (MCS) 能够保证从多个源节点来的数据在所有的节点按相同的顺序接收。这可以在 MCS 内规定适当的优先级和为不同类型的数据确定选路的级别来实现。MCS 支持的种类包括一对一、一对多和多对一路径, 它也可以在多种类型的网络和连接上操作。

(c) T.124 普通会议控制 (GCC) 可以用指定某种业务和管理多点会议来提供会议能力。这种能力包括建立和终止会议, 多种应用中的协商能力和提供通用会议的管理能力, 如

参加人员的清单，主持人控制功能等。GCC 也装备了安全机制，如口令编码保护，能限制未被邀请的人参加会议。

T.120 系列标准为应用开发提供了规范，使不同的厂家的产品和不同的业务能够互相通信。ITU-T 在 T.120 规定了两个高层的规范。一个是 T.127，它为用户提供同时初始化多点文件传输能力。另一个是 T.126，它允许用户在多点文件会议中查阅图像和对它作注释、共享应用程序和交换传真图像。共享可以发生在运行不同操作系统的应用中。

T.120 的主要部分和 ITU-T 的 T.124、T.127 等正式建议一起在 1995 年 3 月的日内瓦会议上得到了批准（表 9-2）。

表 9-2 T 系列国际标准

建 议	内 容
T.120	多媒体数据传输规程
T.122	声像会议多点通信业务规程
T.123	声像会议的通信规程栈
T.124	通用会议控制
T.126	允许用户在多点文件会议中查阅图像或对它作注释、共享应用和交换传真图像
T.127	为用户提供同时初始化多点文件传输能力

表 9-3 描述了 G.7XX 语音压缩编码标准。

表 9-3 G.7XX 语音压缩编码标准

标 准	内 容	码 率	带 宽	算 法	其 他
G.711	语音频带的脉冲编码调制 (PCM)	64Kbit/s	300~3400HZ	PCM	长途电话的语音质量
G.722	7KHZ 频段的 64Kbit/s 语音编码	48、56 64Kbit/s	50~7000HZ	SB-ADPCM	可传送高质量语音和简单的音乐
G.728	16Kbit/s LD-CELP 语音编码	16Kbit/s	300~3400HZ	LD-CELP	ISDN 网传输设备必备

5) PCS 标准。由于基于微机的桌面会议系统日益增多，由 Intel、AT&T、Lotus、HP、DEC 和另外 11 个主流软硬件公司以及 96 个计算机和通信公司联合成立了一个个人会议工作组，简称 PCWG (Personal Conferencing Work Group)。它于 1994 年制定了一个个人会议标准 PCS (Personal Conferencing Specification)，适合于任何网络（数字、模拟、LAN 或 WAN）。此标准包括 PCS'S T120、ITU-T 桌面系统多点电视会议视频压缩协议。与 H.320 不同，PCS 是专为个人计算机制定的，并与各种个人计算机标准兼容。PCS 标准目前仅供 PCWG 成员使用，很快将提供给公众。满足此标准的个人会议系统用户可以无间隙地与其他个人会议系统用户通信。

9.1.2 综合业务多媒体终端的设计和实现

综合业务多媒体通信终端可以广泛用于多媒体通信、交互式分布多媒体系统，特别是各种计算机支持的协同工作系统 (CSCW) 的通用设备，是目前世界各国大力研究的热点课题。由于各种技术发展较快，因此要求总体结构设计要考虑通用性和可扩展性。1996 年 3 月清

华大学计算机系设计实现了一种综合业务多媒体通信终端（IMCT），它的方案是：终端的软、硬件支撑环境采用多媒体个人计算机（MPC）与清华大学和中国银河公司共同设计制造的满足 H.320 系列标准的视频音频实时压缩和处理卡 VC-1000A；系统应用软件采用模块化结构，如视频讨论、多媒体电子邮件、协同编著等，可相对独立运行，又能集成在一个统一的平台上。

1. VC-1000A 的结构和工作原理

VC-1000A 是综合业务多媒体终端的关键部件，它由下述三块卡组成（如图 9-6 所示），三块卡在功能上既有联系，又相互独立。VC-9800 完成视频和音频信号的采集、显示、编辑等功能，VC-9810 完成压缩、解压缩、视音频复合 MVIP 传输等功能，VC-9820 将符合 H.320 标准的压缩码流经线路接口输出至通信网络。

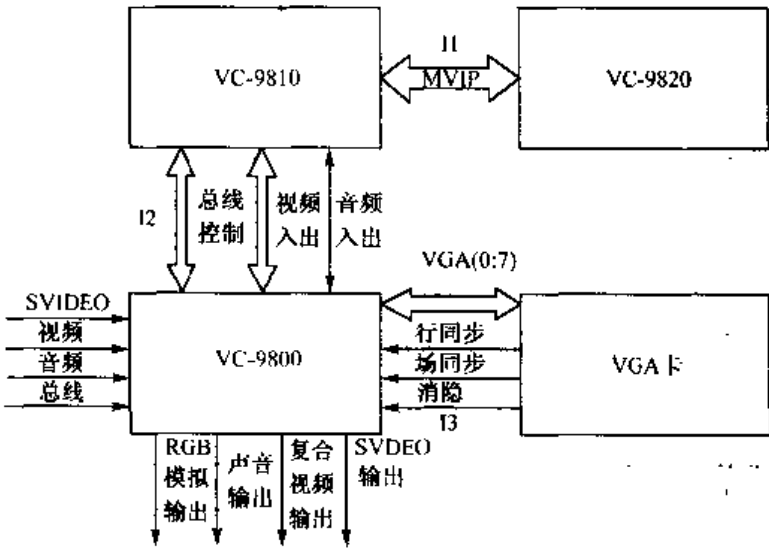


图 9-6 VC-1000A 整体框图

I1MVIP 接口为 40 扁平线 I2: 插针联接方式 I3: VIDEO FEATURE 26 扁平线

(1) VC-9800 视频音频信号采集和处理卡。VC-9800 本身就是一个通用的多媒体处理硬件平台，可完成音视频信号的采集、显示、编辑等功能。图 9-7 是 VC-9800 的原理框图。VC-9800 可分为以下几个部分：

1) 视频输入前、后级。设计采用 Brooktree 公司的 Bt812 作为视频输入前级芯片，Bt812 图像数字化芯片不仅可以接收 PAL 制、NTSC 制输入而且可以接收 S-VIDEO Y/C 分离信号输入，针对不同的使用情形，Bt812 可以采用不同采样频率对输入视频信号进行采样，既可以按像素采样（NTSC640×480×60 场/秒，PAL 768×576×50 场/秒）也可按 CCIR601 标准进行采样，Bt812 支持 24RGB、16RGB，及 4:2:2 16YCrCb 格式，默认采用 4:2:2 16YCrCb 格式，即 8:4:4。在 VC-1000A 设计中，Bt812 输出至 PX2070 的 V1 口，作为本地视频图像输入。

VC-9800 还具有模拟视频输出的功能，这对于视频编辑等场合是十分有用的。特殊视频效果的模拟输出因此成为可能，精心制作的多媒体著作或特殊效果可以显示在电视或录像带上。当背板 VC-9810 与 VC-9800 结合使用时，模拟视频输出图像为远地视频，当 VC-9800 单独使用时模拟视频输出图像为 PX2070 多路视频图像，图形实时叠加输出。在设计中，使

用 Bt858 芯片将数字视频信号转化为模拟 Y/C 供 S-VIDEO 端子输出或复合模拟视频输出，输出格式可以为 PAL 制或 NTSC 制。

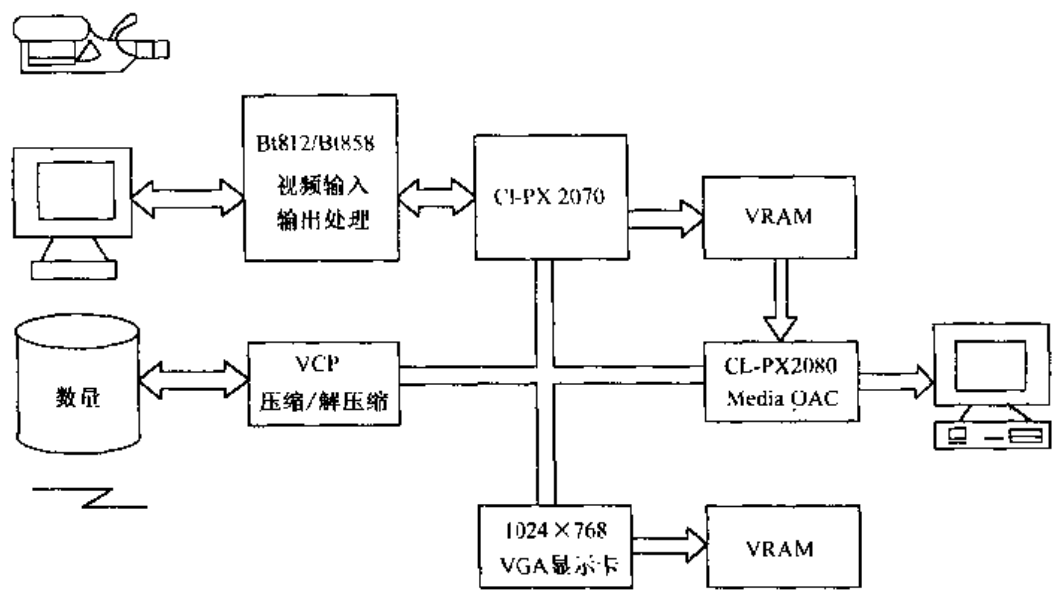


图 9-7 VC-9800 原理框图

2) 图像存储缓冲区部分。为了提高性能，VC-9800 采用 VRAM 作为图像存储缓冲。VRAM 有一个访问端口，一个端口是与 DRAM 相同的随机访问端口，另一个端口是串行存取端口，串行存取端口的串行数据流持续稳定地把数据送至 Media DAC 而不影响 VRAM 的随机存取性能，也不占随机存取的带宽。因此同一个存取周期可实现更高速率的数据交换，实现更高性能。

设计中，选用 512×32VRAM，存储器堆分为两个体，每个体由 8 片 256K×4VRAM 构成，低端体由 RAS0\*控制，高端体由 RAS1\*控制，CAS0\*和 CAS1\*分别存取高 16 位和低 16 位。SCLOCK（串行移位时钟），DTE\*（数据传送始能），SOE[0:1]（串行输出始能）控制串行数据的输出。

3) CL-PX2070 数字视频处理器。CL-PX2070 由四个功能部分组成，核心视频处理单元 VPU 以及三个完成复杂交互功能子单元：视频总线单元 VBU，主机接口单元 HIU，参考帧单元 RFU。

4) PX2080 部分。PX2080 是一个多源，D/A 视频转换器。它能够管理、混合两路不同色度空间及解析度的视频流，PX2080 有三路输入：一路视频端口，一路 8 位 VGA 数据端口及一路 32 位高析度图形数据端口。在设计中视频端口与 VRAM 的串行输出口相连。8 位 VGA 数据端口与 VGA 卡相连。

PX2080 可完成色度空间转换、比例缩放、色度补偿以及图像灰度校正功能。在高性能要求应用领域中，例如图形扫描、图形艺术等要求显示颜色十分精确，在这种情况下，颜色查找表的线性与电子枪之间的非线性偏差不能被忽略，PX2080 通过特殊的图像灰度校正查找表对 DAC 输出进行非线性补偿，使显示在 CRT 上的最终颜色符合线性标准。

此外为避免视频或图形数据的丢失，PX2080 采用同 PX2070 类似的存储器控制策略，

通过 FIFOFULL\*来实现, 当 FIFOFULL\*有效时, 表明 PX2080 256 个双像素大小的视频输入 FIFO 已满, VRAM 可暂缓向 Media DAC 送入数据。

5) 总线控制部分。总线控制主要包含地址译码、地址锁存及生成控制信号等功能, 这一部分电路与通常的总体控制思路没有什么不同, 因此就不加以详细介绍了。

(2) VC-9810 编解码卡。ITU H.320 标准是 ITU 针对电视会议制订的一系列标准的总称。其中包括视频编解码应满足的 H.261 标准, 音/视/用户数据分别复接应满足的 H.221 标准以及帧同步控制和指令应满足的 H.242 标准等。

编解码器的主要功能包括编码(对收发图像、声音信息的码流进行高倍数的压缩, 以适应数字信道数据流的传输速率)、解码(将对方传来的图像、语音按一定的标准还原)、同时为满足点对点通信的要求, 编解码器还包括相互的互通规程以及为实现规程的信令部分。

图 9-8 为 VCP 板的设计框图。本地视频数据经 VC-9800 底板由 VIDEO IN 总线进入 VCP, 声音压缩数据经串行总线进入 VCP, VCP 芯片在微码的控制下按照 H.261 标准对视频流进行压缩, 然后按照 H.221 标准进行复接及向前纠错, 符合 H.221 帧结构的压缩数据流经 TDM 总线送至网络接口板 VC-9820。与此同时, 由网络接口板经 TDM 总线接收到的 H.221 帧结构压缩数据流在 VCP 内部进行分接、解压缩, 视频数据经 VIDEO OUT 送至底板显示, 音频数据送往音箱输出。

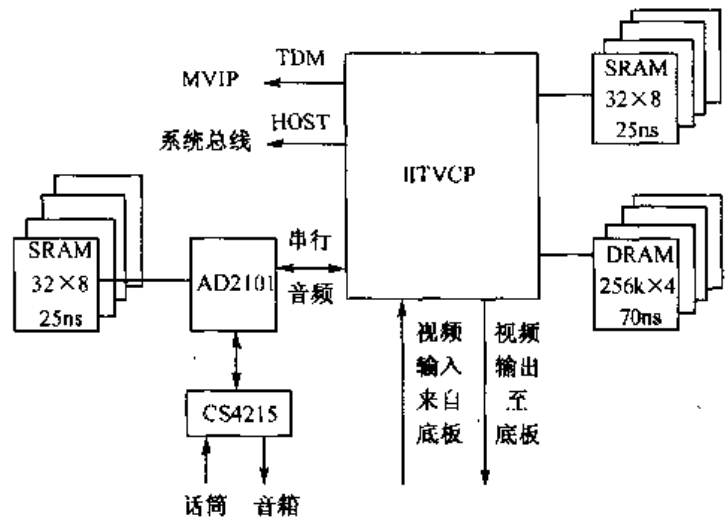


图 9-8 VC-9810 系统简图

VC-9810 主要分为几个部分:

1) DRAM 缓冲区。DRAM 用于存储未压缩图像, 参考帧以及中间数据。DRAM 按 YCrCb 格式组织数据, 编解码进行时, 解码后图像以及编码所需的参考帧及中间处理结果也存储在 DRAM 中。

DRAM 分为两个体, 所有控制信号来自 VCP。DCASO 用来区分高低两个体, 每个体由两片 256K×16DRAM 组成, 数据带宽 32 位。

2) SRAM 缓冲区。SRAM 速度快, 无需刷新, 因而用于存储 RISC 代码, 堆栈及数据, 同时作为 TDM、声音、主机及 H.221 的内部缓冲区。SRAM 同样由 VCP 控制, VCP 与 SRAM 数据带宽为 32 位, 因此设计中把四片 128K×8 组合使用。VCP 一次可读取 32 位, 16 位或 8 位数据, 访问模式由 LCE[3...0]控制。

3) 声音部分。音频编码器由 AD2101 及 CS4215 来实现。其中 CS4215 是一个串行控制的 A/D/A 转换器, AD2101 是定点运算通用 DSP 芯片。

整个音频编解码电路由 AD2101、CS4215、三片 32KSRAM, 一片 256KEPROM 及时钟发生器和外围芯片组成。音频编解码电路与 VCP 相对独立, VCP 芯片不进行音频编解码, 只进行 H.221 的分复接, 所有音频数据的 A/D, D/A 转换, 音频编解码均由本部分电路完成。

4) TDM 总线接口部分。VCP 芯片经过 TDM 总线接口与网络接口板相连, TDM 是一时分复用总线。

5) VCP 功能。RISCIIT 微处理器是 VCP 的核心部件和总控单元, VCP 芯片内部的时序及数据流向均由 RISCIIT 管理, 片内两条总线 SRAM, DRAM 总线分别传送压缩及未压缩数据, 片内数据传递全部采用 DMA, 10 路 DMA 传送的控制与配合由 RISCIIT 运行相应程序进行管理, 至于缓冲区控制策略, 外部接口控制, 寄存器变量访问, 主机交互等功能均由 RISCIIT 实现。

(3) VC9820 接口卡。简而言之, VC-9820 的功能就是把 MVIP 接口信号转换为所需的网络接口信号, 目前包括 RS-449、G.703 规格, 至于 V.35, ISDN 等均大同小异。

1) RS-449 接口板。实际设计中, 采用 ALTERA 公司的 FPGA 可编程逻辑芯片进行设计, 这样做, 一是在设计和布线上较为方便、简洁; 二是软件的模拟、仿真功能较强, 方便设计和检查。设计中, MVIP 接口的时钟与帧同步信号由 VC-9820 提供。

2) G.703 接口板技术指标:

- 线路编码: HDB3
- 线路速率: 2.048Mbit/s
- 数据传输速率: 64Kbits~2.048Mbit/s
- 校验格式: CRC-4
- 传输方式: 全双工通信
- 线路接口: G.703 (E1) 75  $\Omega$  非平衡/120  $\Omega$  平衡  
可软件设置 G.704 帧结构接口  
支持帧同步和复帧同步结构
- 数据接口: MVIP、PC 总线
- 应用领域: 实时图像传输系统 (MPEG、JPEG)  
320 视频会议系统  
高速数据处理系统

功能特性:

VC-9820 卡上的 G.703 (E1) 线路接口与计算机外部的各种高速数据传输设备上的 G.703 (E1) 线路接口连接, 可将计算机系统内的图像、声音、文件数据以 2Mbit/s 或  $P \times 64Kbit/s$  的传输速率, 通过与高速数据传输设备连接的光纤或同轴电缆等高速通信线路进行传输。

## 2. 综合业务多媒体终端的软件

(1) VC-1000A 驱动程序。VC-1000A 驱动程序软件运行平台为 Windows3.1 以上的环境, 采用分层式, 结构化的设计思想, 用 Visual C++ 语言编程实现。主要有下述几个模块:

- 视频信号编解码模块。

- 视频开窗显示控制模块。
- 视频信号数字化处理控制模块。
- 声音压缩编解码模块。
- DSP 2101/2107 Boot 代码模块。
- DSP VCP 通信协议实现模块。
- VCP 图像通信微代码模块。
- Windows 总控模块。

它们可以分成两类：一类是以 IIT VCP 芯片为核心的视频音频压缩编解码及符合 H.320 系列标准的其他软件模块；另一类是视频信号数字化、处理及编辑驱动软件模块。

IIT VCP 芯片是 VC-9810 编解码卡的主要芯片，它采用微码结构，通过内存变量的读写、流输入和输出、警告、监控和执行程序等方式对 VCP 芯片进行初始化，控制执行 H.320 的编解码及其他标准模块。

全局变量既可以通过主机访问，也可以通过 VCP Risc 访问，主机正是通过设置和监控这些变量来控制 H.320 的进程。内存变量的读写过程是访问这些变量的方式，这些变量不仅包括字节、字和双字，同时也可以使用指针和多重访问来访问数组和结构，但是有些变量是只读的，无法通过主机接口改变其数值。

利用流输入输出机制，VCP 芯片的软件编程接口支持连续流或存储文件的操作。无论主机还是 VCP 均可打开、读写这些流，在 VCP 和主机之间通过流操作传送用户数据。

主机可以通过内存读写得知 VCP 状态，同时，VCP 还可以利用告警机制通知主机异常事件。例如，如果解码器在 CIF 和 QCIF 格式切换过程中接收到 PTYPE 标志，将向主机发出告警信号。

主机通过主机接口与 VCP 联系，它包括中断方式、DMA 方式以及寄存器操作方式。VCP 的驱动程序是存放在 VCP 的 SRAM 中。

视频信号数字化、处理及编辑驱动模块是通过 VC-9800 卡实现的。它是介于操作系统和硬件平台之间。为了易于维护和扩展，我们也采用层次结构。第一层与硬件平台有关而与操作系统无关，支持硬件的最基本操作，称为增强模式虚拟设备驱动，它支持中断和 DMA 操作，之上还有一个动态连接程序 (.DLL)，使硬件在虚拟机条件下更好地发挥作用。第二层是结构层和执行层 (construct layer and executive layer)。通过一个例子说明它们的功能，应用程序要完成一个对象在参考帧空间的定位，因此应用程序把直角坐标参数 x、y 传递给执行子层，执行子层判断参数的有效性后，将参数转化成线性地址，送到结构子层，它把与对象有关的数据收集整理，转化为寄存器值，传递给另一层，经过硬件执行，达到应用程序所要完成的任务。

程序流程图如图 9-9 所示。要建立一个数据流单元 DFU (Data Flower Unit)，首先要初始化所用的视频端口，建立视频设备上下文 VDC (Video Device Context)，一个 Windows 程序在写显示内存之前，必须先分配给设备上下文一个句柄，与此类似，在设置 CL-PX2070 数字视频处理器和 CL-PX2080 数字显示处理器时，应用程序也必须先分配 VDC，VDC 负责管理视频操作所需的所有数据流单元，它与一个显示窗口相对应；将数据流单元 DFU 加入到视频设备上下文 VDC，每一个 DFU 对应一个数据，在一个 VDC 中可以包含多个 DFU；设置参考帧 RFU (Reference Frame Unit)，这一步设置对象在参考帧中的位置，显示窗口

在参考帧中的位置及对应屏幕的显示位置，另外还有关于对象和显示窗口的其他特性，如隔行/逐行显示，镜像显示等；设置流参数，流参数用来设置 DFU 使用的资源，包括视频端口，视频处理单元，是否使用 ALU 以及操作时间等；初始化 DFU 的项目控制参数，项目控制是指设置与输入输出有关的视频处理单元，描述视频信号尺寸，裁剪尺寸等；设置 ALU，如果应用程序需要设置 ALU，对视频数据进行算术逻辑运算，完成视频特技处理则应设置 ALU，否则直接进入下一步；启动实时视频，将视频数据显示在计算机屏幕上；实时改变显示参数，如果应用程序想对显示图像窗口的大小，位置及其他属性做修改，可以实时变动设定参数；最后停止实时视频，进入下一种操作。

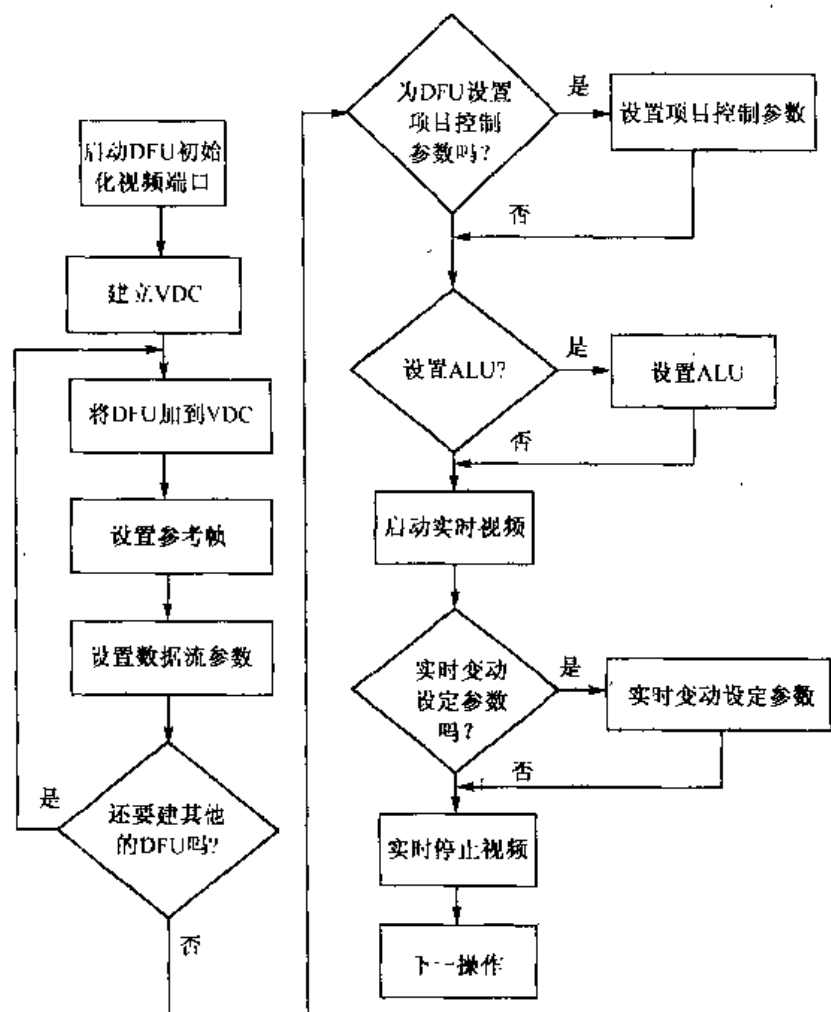


图 9-9 程序流程图

(2) 终端系统应用软件。终端系统应用软件的结构如图 9-10 所示，它包括视频讨论模块，分布式协同编著模块以及多媒体电子邮件模块，各模块可相对独立运行，并集成在一个统一的支撑平台上。

实时视频讨论模块是一个基于分布式平等原则的系统，每个用户自主地决定是否将自己的视频、音频等数据向网络上传送，同时自由选择接收并可以在一定范围内调节其他用户的输出信息，另外通过呼叫和附言等手段，可以实现一个视频讨论系统所必需的功能。实时视频讨论模块的性能如下：



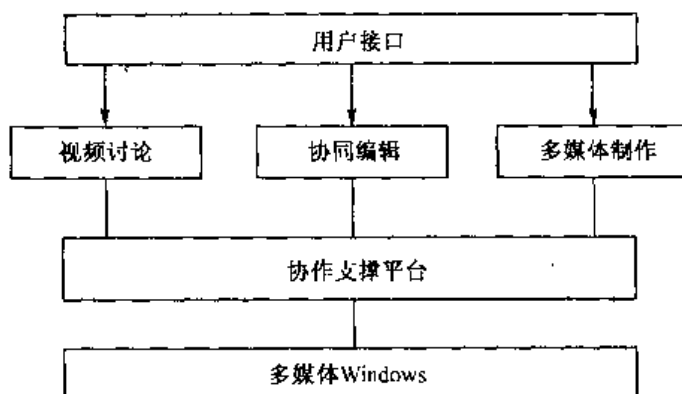


图 9-10 终端系统应用软件结构

- 视频编码符合 H.261 标准。
- 视频实时双工压缩解压缩，压缩与解压缩可分别启停。
- 视频编码比特率范围（64~384）Kbit/s，本地远程均可调节。
- 输入输出为 NTSC/PAL 制全电视信号。
- 以画中画形式同时显示本地、远程视频图像，本地图像位置可调。
- 任意选择远程视频图像源。
- 语音有发言、收听两种状态可任意选择，任意选择收听对象，具有音量调节功能。
- 呼叫（广播）功能，同时可以附加传输文字。
- 网络环境为 Novell 局域网，IPX 协议，具有网络查询功能，给出网上现有的视频、音频数据。
- 方便的帮助功能。

分布式协同编著模块将协同编著的每个站点的屏幕分为三个区域：共享窗口、私有窗口和成员状态窗口。共享窗口提供协同工作者共同讨论的场所，大家可以对同一多媒体文档进行评论和批注，每人可以有一种不同的色笔。私人窗口供用户本人进行私人的编辑操作，也可修改，浏览，查询多媒体著作。成员状态窗口显示所有协作成员的姓名、身份、权力等。

系统中协作成员的社会角色按其在共享显示区的权力判别可分为四类：

- 主编。负责协作成员的加入和退出，发言权管理、批注权管理等。主编是当然的编辑，享有当然的批注权，并且在一次协同工作中不改变。
- 编辑。可以在共享区进行批注和编辑，但主编在必要时可取消其编辑权。
- 发言者。用户申请发言得到主编批准后即自动成为发言者。他具有对共享窗口的编辑权。在协作过程中，系统实时发送发言者的图像和语音给各个协作成员。发言者可以激活窗口中的语音和动态视频数据对象，在系统所有站点进行播放。
- 读者。只能对共享窗口进行浏览。

四种社会角色除主编外都可以动态改变。

协同编著模块的主要功能如下：

- 协作管理菜单。包括协作开始、结束、加入、退出、申请发言权、释放发言权、夺取发言权等。
- 著作管理菜单。包括创建、打开、关闭、退出、按章查询、按页查询、删除一页、

提交一页等。

- 编辑菜单。包括剪贴、复制、粘贴、删除、嵌入新对象、对象操作等。
- 批注菜单。包括批注画笔、批注字符、广告牌等。
- 对象菜单。提供对象特性的操作，如改变对象线颜色，改变对象填充颜色，在三维坐标系统中将选取对象移到其他对象前面或后面，以及将对象前移或后移等。
- 著作库页文档菜单。包括打开一页，共享讨论、更新讨论、清除批注、白板交互等。

多媒体电子邮件模块具体软件结构如图 9-11 所示，总体可以分成两大部分：本地功能和外部程序接口。

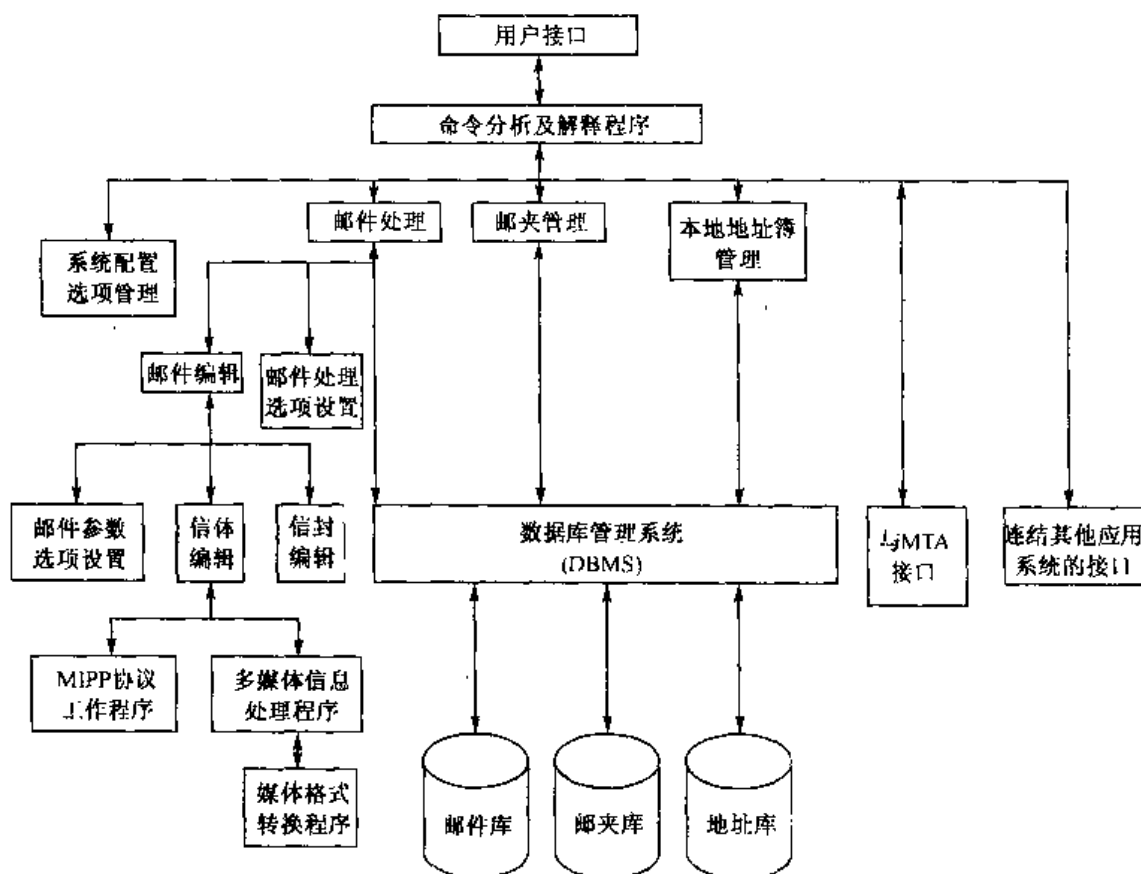


图 9-11 多媒体电子邮件软件模块结构图

本地功能为方便用户处理邮件提供了有力手段，它主要由三部分组成：邮件处理、邮夹管理和本地地址簿管理。

邮件处理包括对邮件的接收、编辑、阅读、转发、回信、发信、打印、输出到外部文件、从外部文件输入等功能。其中信件的编辑又是最基础的，包括信封编辑、邮件选项设置及信体的编辑。信体编辑利用了多媒体信息处理程序来录入及维护信体的各种媒体的信息，并在它们之间建立时间和空间上的联系，由 MIPP (Multimedia Information Processing Protocol) 协议工作程序编码后提交给邮件编辑程序，邮件的阅读则要对信体部分利用 MIPP 协议工作程序解码后由多媒体信息处理模块的演示程序播放。

邮夹是管理邮件的基本单元，每一个邮件都有其归属的一个或多个邮夹。用户可以生成、删除邮夹，并在邮夹中寻找和排序所需的邮件，也可以在邮夹中对邮件实现复制、删除、增

加等功能。邮夹是一个递归的概念。它可以有自己的子邮夹。TH-UA 的用户至少有三个邮夹：发送邮夹、接收邮夹和废纸篓邮夹。

本地地址簿管理用来方便用户编辑信封及搜集参数。从软件实现的角度来考虑，地址簿和邮夹具有类似的结构，这种结构允许组地址的存在。如果有一个工作组经常需召开会议，想利用邮件系统来发出通知，则只需创建一个组地址，把所有成员的地址加入，然后每次发出邮件通知时只需对收件人填入组地址名，则该工作组所有的成员都会收到召开会议的通知邮件。

系统配置选项是指系统初始化后的一些可变设置，如当邮件新到时，是发出一段声音或者是给出一幅生动的动画，或者是给出一段字幕。

数据库则包括邮件库、邮夹库、地址库等，其中邮件库中存储有邮夹的关键字，以及使邮夹库和邮件库建立逻辑联系，三库均有统一的数据管理系统并有统一的接口。

邮夹管理、邮件处理、本地地址簿管理相互之间不独立，而是密切相关，在功能上相互开放，以满足用户对灵活性的需要。

多媒体电子邮件的外部通信协议，是在 X.400 系列建议之上增加多媒体信息处理模块 MIPS (Multimedia Information Processing System) 及多媒体信息处理协议 MIPP 而形成的。CCITT 于 1984 年正式推出了信报处理系统 MHS (Message Handling System) X.400 系列建议，MHS 是一种新的分布式电子邮件系统，它是在模拟邮政系统的基础上，吸取电报、电传、传真技术和计算机信息处理技术的精华，利用存储转发的通信机制，在开放系统互连国际标准模型中形成，MHS 既可以传送文本类型的信息，又可以传送声音、二进制数据、传真和图文数据，还可以对信文加密解密。利用数字签名技术可以对信报的收发者进行严密的身份鉴别。从 OSI (Open System Interconnection) 的基本参考模型来看 MHS 是属于应用层，它本身又可分为三层：用户代理 (UA)，邮件传输代理 (MTA) 和实时传输系统 (RTS)。

UA 提供了信件的编辑、发送、接收、阅读、归档、打印、查询、转发、自动回信、删除等功能，特权 UA 还可以用来管理 MTA 乃至整个邮件系统。UA 假设 MTA 能可靠地传输各种邮件，UA 间能相互理解对方信文内容的基础是它们都按 P2 协议对邮件进行编码和解码，MTA 实现对邮件的存储转发和对用户的管理。MTA 接收邮件并确定其最终去向，并将附加控制信息增加到 UA 递交的数据单元中并发送它。这些控制信息和数据单元按 P1 协议处理，以便各 MTA 之间能相互理解和正确传送邮件。在大多数情况下，MTA 是全时服务的，其本地进程将邮件传送给 UA、MTA 调用下层网络协议 RTS 来打开和关闭网络，以便传送和接收邮件。RTS 则对 MTA 递交的数据单元进行编码解码及分段合段，并借助于会话层服务实现可靠的传输，也可以直接利用更下层的服务。UA、MTA、RTSA 对于普通的文本邮件或许已经足够了，但邮件的内容中出现多媒体后又遇到了一些困难。

这主要体现在：

- 媒体之间演示时的时间和空间关系需要得到体现。
- 多媒体信息与设备相关，而邮件系统中各个 UA 的本地设备可能不一样。
- 应允许扩展新媒体的可能性。

鉴此，我们专门建立了多媒体信息处理模块 MIPS 来处理上述问题，如图 9-12。MIPS 被 UA 作为本地功能调用，MIPS 有其独特的多媒体数据处理模型及协议 MIPP，这些模型及协议能较好地处理多媒体邮件的可共享问题。采取上述折衷方法即扩充用户代理的本地功

能来实现对多媒体电子邮件的处理是合理的。这种方法的优越性体现在以下方面：

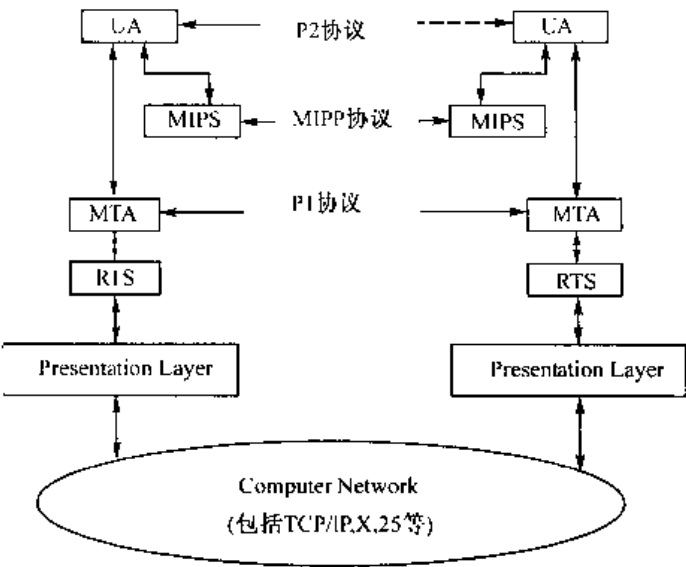


图 9-12 多媒体电子邮件层次模型与协议模型

- 可以充分利用现有的非常成熟的电子邮件系统。
- 随着多媒体技术的进展，通过较小的对本地 MIPS 层的代码修改代价即可将最新技术融入已有的系统中，保证了良好的可扩充性。
- 在单独的多媒体邮件标准出现以前，每一个多媒体邮件团体均可按自己倾向的方式去构造多媒体电子邮件。

值得注意的是：MIPS 和 MIPP 的引入将彻底改变邮件内容的阅读方式，传统的文本邮件可通过编辑程序随意浏览，而多媒体电子邮件的阅读则由 MIPS 中的演示程序按编辑邮件时制定的流程播放。当然，用户可以用播放程序中的暂停、快进、快退、重播等操作来即兴发挥。

9.1.3 多点控制单元 MCU

传统的电话业务仅限于点对点的通信，为了支持位于不同地点的多个成员协同工作，有必要把多个地点连接起来。为此需要多点控制单元 MCU，也可称为桥接设备，来提供这种功能。MCU 是视频会议系统中的关键设备，它的作用是图像、语音、数据信号进行切换，而且是对数据流进行切换，并不像电话交换机对模拟信号进行切换。我们设计的工作速率为 64Kbit/s~2Mbit/s，每次会议工作在一个速率上，如果与它连接终端速率不一致，它便自动地工作在几个终端的最低速率上，在同一次会议，所有终端都会选用同一速率。

MCU 对视频信号采用直接分配的方式，若某会场有发言者，则他的图像信号便会传递到 MCU，MCU 将其切换到它所连接的所有会场。对数据信号，MCU 采用广播方式将某一个会场的数据切换到其他的会场。对语音信号可以分两种情况：如只有一个会场发言，MCU 将其切换到其他所有会场；若同时有几个会场发言，MCU 将它们的音频信号进行混合处理，挑出电平最高的音频信号，然后切换到除该会场外的所有其他会场。

目前 MCU 产品，一个 MCU 最多可连接 12 个会议终端（或别的 MCU）。ITU-T H.320 系列建议的 H.231 和 H.243 分别描述了 2Mbit/s 以下数字信道的视听系统 MCU 的结构组成及利用 MCU 在 2Mbit/s 信道上对多个视听终端建立通信的方法。

## 1. MCU 的结构原理

MCU 的结构原理如图 9-13 所示。

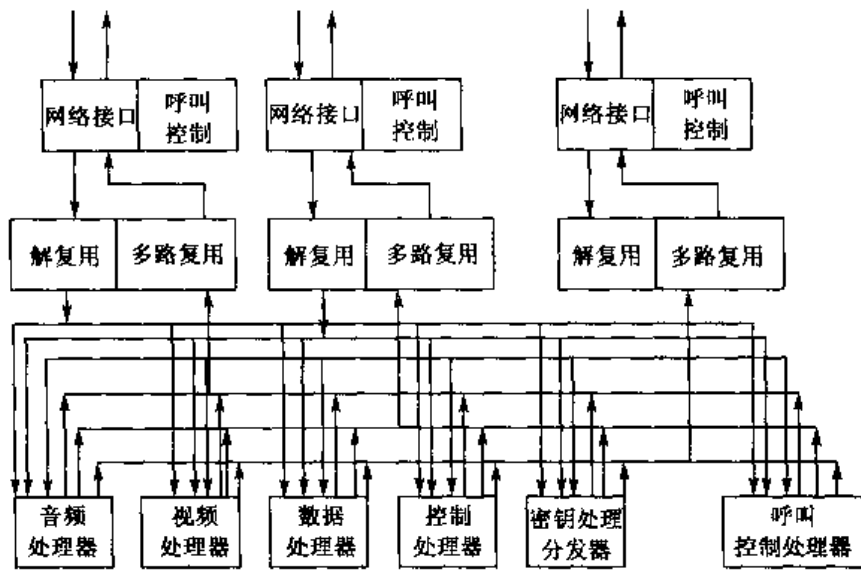


图 9-13 MCU 结构原理图

MCU 的每个端口都连接着一个与呼叫控制相连的网络接口单元，在网络接口单元的 MCU 这一边，数据流包含在按 H.221 建议列出的传输率传输的一个或多个双向通道内。引入数据流进入解复用器内，解复用器将引入数据流分解成视频、音频、数据和控制等几种类型的信息并分别把它们传递到相应的处理器内。多路复用器则将各个处理器送出的数据流组合成一个引出数据流以供传输给会议终端。

(1) 网络接口单元。网络接口单元分输入、输出方向，该单元校正输入数据流中由 H.221 建议定义的 FAS 信号和输出由 H.221、H.230 定义的输出码，并按本系统的时钟定位输入的数据流。在接口模块的输出方向插入所需的 BAS 码和相关信令，形成信道帧，以便输出到数字信道。一个网络接口单元可支持多个逻辑端口。

(2) 端口。端口是一个可支持一个语音或视听终端的逻辑实体，它只与一对多路复用器/解复用器相连。

(3) 解复用器。进入解复用器的信号是会议终端传送的完全符合 H.221 建议的数据流。解复用器的动作与会议终端接收方的动作类似，包括：

- 帧恢复和帧定界。
- 缓冲、同步及相关多个通道的定序。
- BAS 的分解并把其中相应的某些信息送往控制处理器。
- 加密码的分解及解密。
- 分解音频信息并送往音频处理器。
- 分解视频信息并送往视频处理器。
- 分解数据信息并送往数据处理器。

模式控制 BAS 码及相关的音频、视频和数据信息之间必须保持正确的时序关系。

(4) 音频处理器。音频处理器由语音代码转换器 (ATC) 和语音混合模块组成，用来完成语音的处理。ATC 从各个端口输入的数据流的帧结构中分离出 A 律、 $\mu$  律的语音

信号, 并进行译码, 然后送入混合器进行线性叠加, 最后送入到编码器, 形成合适的编码形式, 插入到输出的数据流中。

混合器送往每个会议终端的信号都是所有其他与会会议终端的信号的和。随着音频信号个数的增加, 回声和噪声也会累积, 所以必须采取回声抑制和降低噪音的措施。

如果只有两个终端参加会议, 则音频处理器的语音混合功能由切换功能代替。考虑到视频信号的时延, 在音频处理器内应对音频信号进行一定的延迟 (不大于 30ms)。

(5) 视频处理器。类似于音频处理器, 视频处理器对输入视频信号也有两种处理方式: 一种是进行视频切换, 以便插入信道帧后分配到各个会场, 无需对视频信号进行处理; 另一种是进行视频混合, 当一个会场需同时看到多个会场的图像 (在一个显示器上同时看到其他所有会场的图像或分屏显示) 时, MCU 可对多路视频信号进行混合处理。这种混合是对多个图像以分屏形式空分复用在一个组合图像。

目前的 MCU 产品只实现了视频切换的功能。

(6) 数据处理器。数据处理器为可选单元, 包括 H.243 建议的数据广播功能, 以及按照 H.200/A270 系列建议的多层协议 MLP, 完成非话信息的处理。

在数据广播功能时, 任一时刻只能接收一路 LSD (低速数据) 和/或一路 HSD (高速数据), 广播给由控制处理器按照接收这种数据的能力决定的相连终端。

在 MLP 数据处理功能时, 数据处理器可执行以下的功能:

- 处理远端信息。
- 传输会议控制信号 (请求/确认信号, 主席控制令牌, 音频/视频切换)。

(7) 控制处理器。控制处理器负责确定正确路由、混合/切换以及传递给每个多路复用器的音频、视频、数据及控制信号的格式和时序关系, 同时具有会议控制功能。

(8) 多路复用器。多路复用器把从音频处理器、视频处理器、数据处理器和控制处理器送来的数据流组帧并插入 BAS 码值。

总之, MCU 将各会议终端送来的信号进行分离, 抽取出音频、视频、数据和信令信号, 分别送到相应的处理单元, 进行音频混合或切换、视频切换、数据广播和确定路由选择、定时和处理会议控制等, 处理后的信号由复用器按 H.221 格式组帧, 然后送往相应的端口输出。

MCU 与终端 MCU 之间的连接方式有星形、哑铃形、MCU 星形及分层结构等。

## 2. MCU 控制下的多点会议建立过程

### (1) 多点会议通信过程的建立:

1) 单 MCU 控制下通信过程的建立。在会议终端数目有限, 业务量不大时, 可由单个 MCU 控制整个会议。会议开始时, 多个会议呼叫是逐次进行的, 因此在建立呼叫的初始阶段, 各个终端是按顺序加入会议的。

第一个加入会议的终端 Ta 向 MCU 发送“通信帧结构信号”, 并接收 MCU 发来的 OF 模式的帧结构信号、“SCM-cap” (所选通信模式-能力) 信号。OF 模式是指带有 SC (FAS、BAS) 子信道的一个 64Kbit/s 时隙, 其中的 56Kbit/s 为具有 G.711 建议的 A 律语音信号; “SCM” 为 MCU 所具有的“所选的通信模式”, 意即对每一种传输速率有不同的视频、语音、数据信号速率的组合模式, 在 H.243 中列举了不同通信模式的例子, 例如在 128Kbit/s 传输速率下, 有多个视频、语音信号的速率的组合模式。一种为 16Kbit/s 速率的语音信号、

1.6Kbit/s 的控制信号 (FAS、BAS) 和 110.4Kbit/s 速率的图像信号。三者加起来的总传输速率为 128Kbit/s, 这是一种通信模式; 也可以是由 48Kbit/s 速率的 G.711 语音信号、1.6Kbit/s 速率的控制信号、78.4Kbit/s 速率的图像信号组成的另一种 128Kbit/s 传输速率的通信模式。对每一种传输速率都有若干种通信模式。对于常用的几种传输速率 (64Kbit/s、128Kbit/s、384Kbit/s、1.5Mbit/s、2.048Mbit/s) 而言, MCU 都配置了它们的通信模式。若某次会议使用一种传输速率, MCU 工作过程为: MCU 发送该速率的 “SCM-cap” 标记给终端, 以便根据终端现时的能力 (性能) 自动选择合适的通信模式; MCU 发送 “OF 模式” 信号, 以便使终端从 FAS 信号中提取 8 位组定位信号; 在发送上述能力标记后, MCU 又给终端发送 MCC 信令 (多点会议命令), 终端接收后调节输出传输速率, 使它等于输入传输速率; 继 MCC 信令后, MCU 再发送 MIZ (多点零通信指示), 意思是告诉该终端, 此时没有其他终端加入会议, 并指出会议呼叫已建立; MCU 搜索已连接的端口的输入帧定位, 在建立帧和复帧定位之后, 存储第一个会议终端的能力 (性能), 并指定它为 Ta, Ta 则处于等待状态。

第二个终端 Tb 加入会议的过程: MCU 向第二个终端传送它的 SCM 能力集和 MCC, 指出会议呼叫已建立。MCU 搜索这个连接上的端口上所输入信号的帧和复帧定位, 并存储第二个终端 Tb 的能力, 当 Ta 和 Tb 都已完成和 MCU 之间的以上动作后, MCU 向 Ta 发送 Cancel-MIZ (取消 MIZ) 信令 Ta 和 Tb 便可通过 MCU 进行正常的通信。注意 MCU 在切换视频时, 应向发送视频的终端传送 VCU (快速刷新请求)。

第三个终端 Tc 加入会议的过程: MCU 向第三个终端传送它的 SCM 能力集和 MCC, 指出会议呼叫已经建立, MCU 搜索与第三个终端相连接的端口上输入信号帧和复帧定位, 存储 Tc 的终端能力后, Tc 便可以加入会议。MCU 在把某个终端 (Ta 或 Tb) 的视频切换给 Tc 时, 要发送 VCU 给 Ta 或 Tb。同样, 当把 Tc 的视频切换给其他终端时, MCU 要向 Tc 发送 VCU 信令。

第四个终端及以后的终端加入会议的过程与第三个终端加入会议的过程一样。

由于单个 MCU 所连的会议终端不多, 所以这种情况下多点会议电视系统的会议控制方式 (切换方式) 适合采用语音控制、强制显像控制模式。

## 2) 多 MCU 之间的互联:

(a) 初始化过程。两个及以上的 MCU 互联时, 执行下列过程:

MCU 之间的每个初始化通道建立后, 每个 MCU 像对一个会议终端那样在通道上发送能力集, 然后是 MCU 命令。各 MCU 通过从另一个 MCU 收到的 MCC 的值识别到那个 MCU 的存在。

(b) 主 MCU 的指定。多个 MCU 相连时, 为了统一控制, 避免冲突, 需要有一个 MCU 被指定为主 MCU 执行下面的过程 1。在三个或多于三个的 MCU 采用星形连接时, 一般主 MCU 应为位于星形中心的 MCU, 因此应在呼叫之前被指定好。但对于哑铃形连接 (只有两个 MCU), 则应由其中一个 MCU 执行下面的过程 2。

过程 1 (呼叫前指定主 MCU): 在这个主 MCU 与另外的 MCU 相连时 (通过收到的 MCC 信令识别出来), 它发送 MIM (多点指示主 MCU 信令), 而忽略可能接收到的作为过程 2 的结果的任何其他 MIM 信令。

过程 2 (呼叫前未指定主 MCU): 若一个 MCU 从它的一个端口接收到 MCC 信令但并

未从该端口接收到 MIM 信令时, 则执行内容决定过程 (即若两个相连的 MCU 同时发送了相互冲突的命令时, 则各 MCU 先不对收到的信令作出反应, 而是发送一个随机 SBE 数, 收到的数比自己发送的数大的 MCU 维持原发送的信令, 而另一个 MCU 必须对它收到的 MCU 作出反应, 若两随机数相同, 则再各发送一个 SBE 随机数, 直到不同为止), 如果结果是此 MCU 发送的随机数比收到的小, 则此 MCU 即自认为是主 MCU 并向其他 MCU 发送 MIM 信令。

三个及以上的 MCU 相连时, 主 MCU 的自动指定尚有待进一步研究。

3) 会议结束。如果会议终端依次退出, 直到只剩下一个终端时, 则 MCU 向该终端发送 MIZ 信令, 声明会议结束。

(2) 视频切换。视频切换有两种过程: 在某些 MCU 中, 视频信号不经过任何处理, 只是执行切换功能; 而另一些 MCU 中, 视频信号在被切换之前经过了处理, 以保证在输入信号的校错成帧时无不连续情况。

(3) 数据广播过程。在多点多媒体会议系统中, 除了视频和音频外, 还有其他数据媒体, 如传真、计算机文件等, 在会议过程中可以分发文件、传真等等。这种数据分发由数据令牌方式控制。持有令牌的终端有权向所有其他终端广播它的数据。以下我们讨论 LSD (低速数据) 令牌的管理。对于 HSD (高速数据) 可以采用几乎相同的过程。

1) 令牌的分配。欲传送 LSD 的终端要先去索取 LSD 令牌。该终端用 TD 表示。TD 向 MCU 发送 {DCA-L, (B)} LSD 令牌的分配, 其中 (B) 是要求的数据传输速率, 如果未得到 MCU 的回答, TD 可以过适当时间后再次请求。

一旦 MCU 从 TD 接收了 {DCA-L, (B)}, MCU 动作如下:

如果它已分配了令牌给另外的终端, 或者已经接收了另一个请求, 或者处在数据信道关闭、模式转换处理期间, 或者请求的数据率不是当前公有的能力集中的一个数据率, 或者 MCU 处于资源耗尽状态, 则 MCU 应答 DCR-L, 拒绝这一请求。

如果它既未分配令牌, 又未接收其他的请求, 又无拒绝条件发生, 则 MCU 完成所需要的动态模式转换, 动态模式转换完成之后, MCU 发送 DIT-L 给 TD, 此时 TD 可以开始发送数据。

2) 令牌的释放和重新分配。TD 完成了数据分发之后, 通过发送 IDS-L 或 DCC-L 释放令牌, MCU 接收到 DIS-L 或 DCC-L 后, 发送 DCR-L 给 TD, 并发送 DCC-L 去关闭所有其他终端与这个 LSD 有关的数据信道, 然后这个 LSD 令牌可以重新分配。

3) 令牌的回收。主席控制终端可以向 MCU 传送 DCR-L 回收令牌, MCU 接收后再向其他终端广播 DCR-L, 强行回收数据令牌, 然后 MCU 关闭 LSD 信道。必要时 (如为了解除某种错误) MCU 可以自选回收令牌。

终端在持有令牌期间接收了 DCR-L, 必须停止在 LSD 信道上的数据传送, 然后它必须传送 DIS-L 或 DCC-L 给 MCU。

(4) MCU 控制模式。在多点会议电视系统进行会议控制时, 可采用语音控制、强制显像控制、主席控制等三种方式。

1) 语音控制模式。语音控制模式为全自动工作模式。

在一个多点会议进行的过程中, 当同时有多个会场要求发言时, MCU 从这些会场终端送来的数据流中提取出音频信号, 在语音处理器中进行电平比较, 选出电平最高的音频信号,



将最响亮的语音发言人的图像与语音信号广播到其他的会场，为了避免不必要的干扰引起的切换，MCU 的切换过程应有一定的时延。切换前的发言时间应为 1~3s。

为避免咳嗽、关门声等其他干扰造成的误切换，两次切换之间的时间间隔应为 1~5s。

2) 强制显像控制模式。强制显像控制模式又称为演讲人控制模式。

当召开一次多点会议时，要发言的人（或称演讲人）通过编解码器向 MCU 请求发言。此时如按桌上的按钮，或触摸控制盘上相应的键，编解码器便给 MCU 一个请求信号 MCV（多点强制显示像命令），若 MCU 认可便将它的图像、语音信号播放到所有与 MCU 相连接的会场终端，同时 MCU 给发言人一个已“播放”的指示 MIV（多点显像指示），使发言人知道他的图像、语音信号已被其他参加会议的会场收到。当发言人讲话完毕时，MCU 将自动恢复到语音控制模式。

需要说明，上述两种控制模式仅适用于参加会议的会场不多的情况。尤其是在语音控制模式下，比较的语音路数越多，则背景噪声越大，MCU 的语音处理器已很难选出最高电平的语音信号。为可靠起见，以一个 MCU 控制十几个会场终端数目为限，再多会场数目不适于采用语音控制。

3) 主席控制模式。主席控制模式符合 H.243 建议。在这种控制模式下，会议主席行使会议的控制权，他掌握行使主席权力的令牌，该令牌为主 MCU 所承认。主席可点名某分会场发言，并与它对话，其他分会场接收他们的图像和语音信号。分会场发言需向主席申请。若主 MCU 收回令牌，则原主席将失去会议主席的权力；他在的会场也就不再是主会场。

### 3. 视频会议系统的控制

T.124 建议（通用会议控制）草案提供了会议管理和对声像与视听终端及 MCU 进行控制的高层框架。该建议与 T.122（用于声像会议和视听会议的多点通信服务 MCS）、T.125（用于声像会议和视听会议的多点通信服务规程）及 T.123（用于声像会议和视听会议应用的网路特定传输规程）一起建立和管理会议控制的机制。

T.124 建议包括下列通用会议控制 GCC（General Conference Control）功能组件：会议建立与结束，管理会议名表，管理应用名表，应用登记服务及会议指挥等。限于篇幅，这里只介绍会议建立与结束部分。

为便于说明，先介绍会议控制的几个关键概念：

会议轮廓（conference profile）：由 GCC 提供者维护的一个数据库，由与会议整体相关的信息如会议名、口令等组成。

会议名表（conference roster）：由 GCC 提供者维护的一个数据库，由组成会议的各结点的相关信息的一个列表组成。

指挥者（conductor）：指挥者是会议内的一个结点，它可控制该会议某些方面的情况（如对会议参加者的控制及对会议结束的控制等）。一个会议可有一个指挥者，或者没有。一个结点通过获取指挥者令牌或通过请求或接收现任指挥者的指挥权成为指挥者。

结点及结点控制器（node, node controller）：结点是一个会议终端或 MCU。一个结点可以同时是一个会议终端及 MCU。结点控制器是控制一个结点的功能实体。

GCC 提供者（GCC Provider）：对一个会议终端或 MCU 上的结点控制器或应用协议实体提供服务的机构。

顶层 GCC 提供者 (Top GCC Provider)：与顶层 MCS 提供者同驻于一个结点的 GCC 提供者。顶层 GCC 提供者对同一会议内的其他 GCC 提供者无任何需求。

(1) 会议轮廓。所有会议都有下列属性，这些属性在会议创建时定义并在整个会议举行期间保持不变，每个结点参加会议时，这些属性都被传递给它。我们把这些属性称为会议的轮廓。

1) 会议名。一个数字字符串及一个可选的文本字符串，标识一个会议。

2) 会议描述符。描述一个会议的一个可选的文本字符串。

3) 口令保护或无口令保护。

4) 可列表的或不可列表的。决定本会议是否列入查询会议时提供的会议列表。

5) 可指挥的或不可指挥的。决定本会议可以置为指挥模式还是始终不可指挥。

6) 结束方式。决定本会议是只要明确结束（手工结束）时就结束，还是直到所有参加者离开时自动结束。

7) 特权列表，一些列表的组合，指示正常情况下只有会议召集者才能拥有的特权中哪些可被指挥者享有，哪些可被指挥模式下的任一结点享有，哪些可被无指挥模式下的任一结点所享有。

(2) 会议建立过程举例。会议可在不同条件下通过不同的方式建立。最简单的会议是点对点会议，在这种情况下，呼叫过程不涉及 MCU。在会议通过一个或多个 MCU 建立的情况下，典型的呼叫方式是相遇方式 (Meet-Me Style)，此时会议的所有参加者都向一个 MCU 拨号呼叫或呼叫输出方式 (Call-out Style)，此时由 MCU 向所有与会者拨号呼叫或呼叫通过方式 (Call-through Style)，此时由会议召集者（终端）呼叫 MCU，然后再由 MCU 呼叫其他与会者。

1) 相遇会议的建立。在一个相遇会议中，会议在一个 MCU 上建立，终端结点（如果需要，还有其他 MCU）向该 MCU 呼叫并参加会议。如果其他 MCU 已参加本会议，终端结点可以呼叫这些 MCU 中的一个已参加会议。

相遇会议的初始创建可在带外完成（例如由 MCU 本身创立），或由第一个呼叫 MCU 的结点创建。在前一种情况下，GCC 不参与创建。在后一种情况下，会议由召集结点在召集结点与该 MCU 之间的连接上发送会议创建请求原语在该 MCU 上创建。不论是哪一种情况，会议创建时所处的结点都成为顶层 GCC 提供者，而召集结点本身（前者为该 MCU，后者为发送创建请求原语的结点）则成为会议召集者，具有特殊的地位。

当使用会议创建请求原语远程创建一个会议时，在请求原语中固定了一个会议名。若该会议名在该 MCU 上已被其他会议使用，则该 MCU 的结点控制器使用一个会议名修改符保证该 MCU 上会议名的唯一性。其他结点若欲参加此会议时，在参加请求原语中必须提供这个会议名修改符。

远程创建一个会议时，创建请求原语也可包括一个可选的召集者标识符。这个标识符只在召集者想与会议断连且想以后重新参加并仍想继续拥有召集者特权时才有用。

欲参加一个会议的结点将在该结点与任一参加该会议的 MCU 之间的连接上发送会议参加请求原语。典型情况下，结点不必预先知道想参加的会议的名字。这时，在参加会议前，结点可查询 MCU 以获得可参加的会议的列表。这是由发送会议查询请求原语完成的。得到的回答是一个会议查询确认原语，该原语表明了该终端所连的结点的类型（即 MCU），并

提供一个该 MCU 现在参加的所有可列出会议的列表。这个列表包括每个会议的会议名，如果需要还有会议名修改符，以及会议的其他特性如会议是否是口令保护的。一旦选择了会议名，即可通过发送一个规定了该会议的会议名（如果需要还有会议名修改符）的会议参加请求原语参加该会议。如果会议需要口令，参加请求原语中也需要包含正确的口令。

在会议名通过带外被通知给结点时，结点可通过在会议参加请求原语中提供会议名直接申请参加会议，而无需先查询可参加的会议。若结点参加会议时需与多个相对独立的 MCU 相连时，由于命名冲突某些 MCU 可能需要会议名修改符，由于不使用会议查询原语时获取会议名修改符可能很困难，故建议在有命名冲突的可能性时，参加一个相遇会议前先使用会议查询原语，并建议在创建一个相遇会议时规定一个会议描述符，这样，具有相同会议名的多个会议更可能通过拥有不同的会议描述符而区分开。但一般情况下，在创建一个相遇会议时，最好选择一个对所有 MCU 都具有唯一性的会议名，而无需使用会议修改符。

2) 呼叫输出会议的建立。在呼叫输出会议的建立过程中，由 MCU 创建一个会议，并由该 MCU 呼叫及邀请与会者参加会议。典型情况下，会议创建时规定了锁定特性，并可能规定为不可列出的。

然后，召集者 MCU 将通过依次邀请每个结点参加会议来与各终端建立物理连接。这是由向被邀请结点发送会议邀请请求原语完成的。由于会议是由该 MCU 创建的，该 MCU 就成为该呼叫输出会议的顶层 GCC 提供者。

3) 呼叫通过会议的建立。除了会议由一个启动终端启动创建外，呼叫通过会议的建立与呼叫输出会议的建立情况是相似的。在这种情况下，启动终端向 MCU 发起连接并使用会议创建请求原语创建会议。像呼叫输出会议一样，典型情况下，呼叫通过会议也是锁定的和不可列出的。呼叫通过会议创建时一般规定为自动结束，即在所有结点断连时会议将自动结束。典型情况下，呼叫通过会议规定一个空会议名，因为不需要别的结点明确地参加会议。如果在召集结点（MCU）上已有一个具有空会议名的会议存在，则结点控制器只简单地选择一个随机名作为新会议的会议名。

4) 点对点会议的建立。点对点会议与以上讨论的几种会议有明显不同，因为它只涉及两个终端结点，而不需要 MCU。会议建立前，呼叫终端先向被呼叫终端发送会议查询请求原语，以获悉对方是终端还是 MCU。在点对点呼叫的情况下，响应查询请求原语的会议查询确认原语表明了对方是否为一个会议终端（或会议终端/MCU 的混合结点）。一旦知道对方是终端，呼叫终端即通过发送会议创建请求原语创建一个新的会议。典型情况下，该会议将被规定为空会议名、锁定的、不可列出的、自动结束的。

### 9.1.4 视频会议系统的服务质量（QoS）及资源管理

视频会议系统是一种分布式多媒体信息管理系统，或称分布式多媒体通信系统。它不仅要求能够快速传送视频、音频和数据，而且要求要能满足一定的服务质量，如视频和音频连续媒体，必须保证在明确规定的时间内无差错的传送给用户，以便在终端系统播放具备良好的质量。为了获得服务质量保证，在业务执行过程中，需要对计算机、网络、MCU 及终端的各种资源进行控制和管理。

视频会议系统需要：

- 高数据吞吐量。视频会议系统中的视频和视频数据的特性需要高的数据吞吐量，即

使经过压缩，一个视频数据流也需要 64Kbit/s 到 2Mbit/s 的数据吞吐率。

- 实时性。视频会议系统的终端同时要播放视频、音频和数据信息，这些信息之间存在较严格的时间要求，因此需要为无差错传送提供时间保证。
- 服务质量保证。用户使用视频会议业务总是和其他业务进行对比，如果不能提供一定的服务质量保证，用户就会选择其他种类的业务，如无线电、电话等。

上述三种需求都表明需要时间、空间和频率资源管理系统以满足视频会议系统对服务质量的要求。为此，在这一节中我们将进一步讲述视频会议系统的服务质量，视频会议系统资源的静态管理，资源的动态管理以及视频会议系统的安全保密问题。

1. 视频会议系统的服务质量 (QoS)

视频会议系统需求是多样化，多媒体业务也是多样化，为使这些需求和业务能够定量化的描述，往往采用参数化，这样不必每个应用都实行一套新的系统服务集。为使服务参数化，国际标准化组织采用服务质量 QoS (Quality of Service)，到 20 世纪 80 年代末，人们开始从不同角度深入研究 QoS，主要集中在 QoS 管理，QoS 标准，QoS 主客观的统一等问题。所谓 QoS 管理，是指计算机和网络系统采用一定的方法，满足用户应用的服务请求，并保证 QoS 的过程。在这一过程中计算机和网络系统将用户应用的服务请求映射成一些预先定义的 QoS 参数，进而与系统和网络的资源对应起来，通过资源的分配和调度，满足用户的应用需求，这一过程叫 QoS 协商。QoS 协商有时可能不成功，原因是系统和网络无法完全满足带宽、延迟和正确率的要求，这时通常又要进行 QoS 的重协商，以确定是否允许降低 QoS。

国际标准化组织 ISO 所制定的著名的七层标准计算机通信协议 OSI-RM 中，定义了一些数据通信应用的，局限于会议层和传输层的一些静态参数，OSI-RM 中将 QoS 参数分成面向功能的 QoS 参数和非面向功能的 QoS 参数两类，如表 9-4 及表 9-5 所示。ISO 定义的这些 QoS 参数，在实际操作中只允许用户应用将高层的 QoS 参数简单不变的映射到低层，而且，为传输层定义的 QoS 参数都是局限于会话的各个阶段的。

表 9-4 OSI 面向功能的 QoS 参数

参 数	意 义
通量	单位时间内在一个连接上传递的最大字节数
传输延迟	从数据传输请求开始到数据传输完成确认为止的时间间隔
出错率	数据单元错传、丢失或重传的概率
建立连接延迟	从请求建立连接开始到建立连接确认为止的时间间隔
连接失败率	建立连接失败的概率
传输失败率	传输失败的概率
重置率	在给定的时间内服务提供者释放连接或重置连接的概率
释放延迟	从释放请求开始到释放请求确认为止的时间间隔

表 9-5 OSI 非面向功能的 QoS 参数

参 数	意 义
访问权限	防止非法用户访问
优先级	包括传输优先级和使用优先级
成本	信息传输时所消耗的资源或资金

在用户层，一个网络多媒体系统可能包含三个抽象层：应用层、系统层（通信和操作系统）和设备层（网络和多媒体终端设备），这三层都需要考虑 QoS 参数。

应用层 QoS 参数描述了应用业务的需求，通常用媒体质量和媒体关系来说明，媒体质量包括源和接收设备的特性，如端对端的延时；媒体关系说明了媒体之间的关系，如媒体之间的同步关系，媒体转换流间和流内之间的关系。

系统层 QoS 参数描述了从应用层 QoS 得到的关于通信和操作系统的需求，这些参数以定量和定性的形式给出，定量的标准有每秒位数、允许误差率、任务处理时间和数据单元大小等；质量判别准则说明期望的业务，如流间同步、按次序的数据传递、错误恢复机制和调度机制等。

网络层 QoS 参数可以用网络负荷及网络特性来说明，网络负荷指正在进行的的需求，如相继到达时间；网络特性如延时、带宽和抖动（多个包之间延时的变化）。网络服务依赖于网络业务量模型（连续请求的个数）和业务参数如峰值数据传输率或突发长度等。

设备 QoS 参数，如媒体的数据传输率等。如果没有明确指明 QoS 参数，则采用最佳服务。下面给出有关多媒体服务通用的 QoS 参数，如表 9-6 所示。

表 9-6 多媒体服务 QoS 参数

媒体类型	业务质量参数	范围	质量特性
音频应用层 QoS	样本大小	8 位	电话语音质量
	采样率	8kHz	中间延时 125 $\mu$ s
	样本大小	16 位	CD 音频
	采样率	44.1kHz	中间延时 22.7 $\mu$ s
	回放点	大约从 100~150ms	取决于网络延时
音频网络层 QoS	端对端延时	0~150ms	绝大多数可接受
		150~400 ms	可能会影响一些应用程序
		400 ms 以上	不可接受
	环绕延时	直至 800 ms	谈话情形可接受
	包丢失	$\leq 10^{-1}$	电话质量
		16Kbit/s	电话语音
		32Kbit/s	音频会议语音
		64Kbit/s	接近 CD 音频
		128Kbit/s	CD 质量音频
视频应用层 QoS	帧率	30fps	NTSC 制式
		25fps	PAL 制式
		60fps	HDTV 格式
	帧宽度	$\leq 720$	MPEG 编码后的视频信号
	帧高度	$\leq 576$	垂直大小
	颜色精度	每像素 8 位	256 色的灰度精度
		每像素 16 位	6536 种可能的颜色
	像素纵横比	4:3	NTSC 和 PAL 电视制式
		16:9	HDTV 格式
	压缩比	2:1	HDTV 无损压缩
		50:1	HDTV 有损压缩

(续)

媒体类型	业务质量参数	范围	质量特性
视频系统层 QoS	解码器缓冲	$\leq 376832$ 个位	与 MPEG 相关的参数
视频网络层 QoS	带宽	$\leq 1.86\text{Mbit/s}$	MPEG 编码的视频
		$64\text{Kbit/s} \sim 2\text{Mbit/s}$	H.261 编码的视频
		$1544\text{Kbit/s} \sim 2\text{Mbit/s}$	H.120
		$140\text{Mbit/s}$	TV, PCM 编码
		超过 $1\text{Gbit/s}$	HDTV 未压缩质量
		$500\text{Mbit/s}$ 左右	HDTV 无损压缩
		$20\text{Mbit/s}$	HDTV 有损压缩
	比特误码率	$\leq 10^{-6}$	长期比特误码率
	包丢失	$\leq 10^{-2}$	未压缩视频
		$\leq 10^{-11}$	压缩视频
	端对端延时	$\approx 250\text{ ms}$	
音频/视频	同步误差	$\pm 80\text{ ms}$	唇同步
音频/图像	同步误差	$\pm 5\text{ ms}$	带有注解的音乐
数据 网络层 QoS	带宽	$0.2 \sim 10\text{Mbit/s}$	文件传输
	端对端延时	$\approx 1\text{ s}$	
	包丢失	$10^{-11}$	

下面简单介绍一下 QoS 的管理和控制:

(1) QoS 的分层、分维和分类 (Layer、Dimension and Category)。早期 QoS 概念是笼统定义的, 没有考虑应用和网络的分层, 不能很好的描述用户 QoS 的请求, OSI 虽然按层次描述 QoS, 但由于制定各层标准的工作小组是单独工作的, 所以各层 QoS 之间没有明确统一的定义。CCITT 的 QoS 只局限在传输层上。最近有人把 QoS 划分为 4 层 (Layer): 应用层、综合层、传输子系统和网络层。每一层又划分成许多维 (Dimension), 维是用来描述系统行为的某一方面的特征, 如吞吐量、延迟、误差率、同步和多模式, 如果说层在服务质量体系中是上下纵向的, 那么维就是左右横向的, 类则是横向一些维的集合, 类作为 QoS 管理的逻辑单元。

(2) QoS 参数的定义和映射 (Specification and Mapping)。QoS 参数指的是能够为系统解释, 用户需要的 QoS 的一些性能指标, QoS 参数的定义除了要包含相应指标的上下界限, 还应包含达不到要求时系统的反应等。QoS 参数的定义在系统各个层次是不同的, 但同一维所对应的不同层次的 QoS 参数含义是相关的。QoS 的映射是指系统自动将用户高层次 QoS 请求解释成较低层次的 QoS 参数, 不必关心该 QoS 请求在较低层次是如何通过各种复杂方式表示的。

(3) QoS 的协商和准许控制 (Negotiation and Admission Control)。QoS 协商包括用户和系统之间的 QoS 协商和端到端 (end to end) 的 QoS 协商。用户和系统之间的 QoS 协商过程如下: 首先分析用户应用需求, 将其分解为一些元素, 使得每一个元素都可由一个独立的 QoS 参数表示, 且这些参数合起来足以满足用户的 QoS 需求, 然后把所得的参数组向下层进行映射, 同时进行交换, 以确定下层是否支持这些 QoS 要求。端到端的 QoS 协商分为前向和后向两个阶段, 前向阶段由一端用户发起, 将其 QoS 请求逐级向下层映射并作比较,

如得到允许,就通过网络发给另一端的用户。在网络另一端,通过逆向映射,到达用户应用层。后向阶段,由网络另一端的应用根据其能力修改得到的 QoS 参数然后再逐层向下协商,并通过网络传回发起端,这时所得到的 QoS 参数就是两端用户最终通信的 QoS。

在 QoS 协商过程中,对于用户应用每一个新的需求,系统都要把其对应的 QoS 与系统当前的资源进行比较,并根据一定的策略确定该行为是否能够实施,这种比较过程和判断策略就是 QoS 准许控制的主要内容。

(4) 资源分配和调度 (Allocation and Scheduling)。QoS 映射的最终的结果是将用户应用的 QoS 请求映射成系统可以控制的资源,这些资源包括计算机系统的 CPU、进程、线程、内存缓冲区、网络的带宽和路由等,系统通过分配和调度这些资源满足用户应用的 QoS 请求。合理地进行资源分配和调度是成功地进行 QoS 管理的关键,解决这个问题需要有效的资源分配和调度算法,如 Internet 中的资源预约算法 (SRP),基于公平队列和静态优先级的算法、分级带宽分配算法及广域网通道建立算法等。

(5) QoS 维护、监视和控制 (Maintenance, Monitoring and Policing)。QoS 的维护是指在网络负载不断变化的情况下,动态地管理资源,以确保 QoS 协商过程中所确定的 QoS 不受负载变化的影响。QoS 监视是指系统的每一层动态地监视下一层的行为是否保证了预期的 QoS。通常采用的维护方法是,采用反馈循环将实际所得到的 QoS 与预期的值作比较,通过调整系统资源来改善 QoS。在最上层的用户应用接口处,当系统检测到下层已无法保证用户的 QoS 要求时,通常应向用户报告 QoS 降级信息,让用户自行决定是适应还是终止,或者对该行为的 QoS 重新协商。QoS 控制是指在的 QoS 监视基础上,对越界行为进行强制性处理。

(6) QoS 再协商 (Re-negotiation)。如果用户设置的初始 QoS 的要求得不到满足,用户修改 QoS 要求,需要重新协商称之为再协商。在传输过程中,若用户对当前服务质量不满意,用户也可重新申请 QoS,这也称之为再协商。QoS 再协商所指的并不是某一个 QoS 元素的局部调整,而是指整个行为所有的 QoS 元素,由用户应用级开始往下重新进行 QoS 协商的过程,QoS 再协商通常用于 QoS 降级后的响应过程。

## 2. 资源的静态管理和动态管理

视频会议系统是一个复杂的计算机网络系统,它包含很多资源,这些资源分布在计算机、终端、多点控制单元、路由器、交换机以及通信信道上。分布在计算机上的资源主要有处理器、总线、内存、进程、线程、网络接口板、磁盘空间等,分布在路由器和交换机上的资源有处理器、总线、缓冲区、各种数据表格、数据接收端口、数据发送端口等。除了上述资源外,还有一些不可见成份,即对这些可见资源的使用策略,它包括内存管理策略、进程和线程调度算法、缓冲算法、差错恢复算法、流控制算法及阻塞控制算法等。也有人把视频会议系统的资源分成动态资源和静态资源,动态资源提供各种业务,如 CPU 或处理各种协议的网络适配卡等;静态资源标志了动态资源所需的系统能力,如主存(缓冲空间)或带宽(链路吞吐量)。资源可以被一个进程独占,有的也可被多个进程共享,例如一个扬声器可以是一个独占的资源,而带宽则可以是一个共享的资源。如果资源在系统中只存在一次,可以将该资源称为单次资源,相反可以称为多次资源,如在一个多处理器系统中,各个 CPU 是多次资源,而在传统的工作站中,CPU 则是单次资源。

在明确了视频会议系统的服务质量要求后,我们就可根据这些要求研究资源管理问题,

找出相应的解决方案。资源是由视频会议系统的资源管理子系统进行管理，它包括主机和网络节点之间的资源管理协议，用它交换有关资源的信息。资源管理的主要目的是保证多媒体数据的传递，主要有下述三个内容：

- 在会议建立过程中保留和分配资源，以保证服务质量能够满足 QoS 的要求。
- 在多媒体数据传送过程中，采用适当的 QoS 服务策略进行资源分配。
- 视频会议的各个阶段要进行相应的资源改变。

(1) 资源的静态管理。在视频会议系统中，终端用户都希望系统能够按用户提出的 QoS 需求提供服务，在多媒体数据发送之前，用户定义的需求必须和资源管理系统进行通信，然后协商 QoS 参数，如果有分歧，在层之间进行说明和解释；最后，所需资源必须在发送方和接收方之间的通路上进行准许、保留和分配。在会议建立过程中完成上述三步，会议结束后，要考虑资源的释放。

1) QoS 的协商和解释。QoS 的协商可是同一层，如应用层对应用层，也可以是不同层，如应用层对系统层，按照 ISO 的术语，对等协商就是呼叫者和被呼叫者的协商，而不同层地协商可以是业务使用者和业务提供者的协商。协商的目的是：

- 在服务的使用者和提供者之间建立公共的 QoS 参数值。
- 任何时刻为满足需要使用必要的资源，以达到充分利用资源的目的。

在视频会议系统中，最常见的 QoS 协商过程是能力交换的过程，在会议系统建立过程中，会议双方为了能够正确通信，必须进行能力的协商，达成一致后，才能进行后续的过程。能力的选择和确定是为了保证只提供和接收对方能够支持的业务，对于点对点的情况，能力交换的发起方必须对通信双方能力集进行分析，求取能力公共集的最大值，并采用该参数进行模式切换，对于多点会议的情况，上述协商过程是由多点控制单元 (MCU) 来完成的。

2) 资源许可。每层经过查询或通过协商和解释得到 QoS 说明后，接下来的一步是资源许可。许可过程基于 QoS 说明并使用一个嵌入在资源管理器中的业务。许可业务检查是放在发送方和接收方之间通路的各个节点上。许可测试取决于在每层的控制机制的测试。同时，任何一个 QoS 协商和资源许可必须和一个开销函数紧密联系在一起，可以决定向一个事先预定服务的用户的收费少于一个需要即时服务的用户。

为了控制资源的有效性，我们在视频会议系统中采用了三种类型的许可测试策略：

- 共享资源的调度测试，例如 CPU 调度性测试和在网络接入点的包调度性测试以及各个网络节点处进行的延时、抖动、吞吐量及可靠性的测试。
- 为保证延时和可靠性对分配的缓冲进行的空间测试。
- 为保证吞吐量在主机总线和网络中进行的链路带宽测试。

3) 资源的保留和分配。资源的保留和分配基于许可测试的结果，在大多数系统中资源的保留和分配是简单的，即资源只在发送方和接收方单方向进行保留。资源的保留和分配需要嵌入在资源管理器中的一套保留和分配的函数以及在资源管理器之间进行信息通信的保留协议。

下面给视频会议系统中所采用的资源保留和分配协议的工作流程：

- 呼叫建立的发起方以一个保留消息（连接请求）发送一个 QoS 说明。
- 在路径中的每个路由器、交换机或其他实体中，保留协议向对应的资源管理器传递一个新资源保留请求。



- 经过许可决策，资源管理器保留资源并为 QoS 预备更新特定的业务信息。
- 在链路的末端，最后一个实体发回一个分配信息（连接确认）给发起方，其格式为：或接受、或修改和或拒绝的回答以及最后的 QoS 值。
- 分配消息通过路径返回到发起方。按照接受、修改或拒绝最后 QoS 参数的消息指令，每个资源管理器轮流分配或释放保留的资源。

4) 资源的释放。经过传输之后，资源被释放。CPU、网络带宽和缓冲区空间都被释放，并且媒体流的连接也被关闭。关闭进程必须在不打断其他网络流的情形下进行并且通过资源管理器更新释放资源为可用资源。

在视频会议系统中，其资源释放有三种方式：

- 发送方请求关闭呼叫。这意味着在发送方和接受方路径上对应的呼叫的所有连接的资源都必须被释放，并且必须对每个节点更新资源的可用性。
- 接受方请求关闭呼叫。该请求发送给发送方，并按照发送方关闭的形式进行资源的释放。
- 资源维护系统指示关闭。这可能发生在，虽然传输已经终止但是资源没有被释放。因而，一个和定时器一道工作的监视子系统应该监视连接状况和检测参与方是否仍然处于活动状态。对于此种情形，资源释放由资源管理系统本身发起。

(2) 资源的动态管理。假设已经完成了 QoS 的协商和资源分配的需求，资源管理器必须支持资源在传输期间的可访问性，即资源的动态管理。满足时间、空间、设备、频率和可靠性需求的工作属于不同的管理部件，如进程管理器、缓冲管理器和流量控制器等部件。

1) 进程管理。多媒体传输涉及多个独立的任务：数据的移动、控制和同步等。由于所有这些任务是由同一个资源管理器进行处理，因而其使用必须进行调度。调度由进程管理器完成，进程管理器是资源管理器的一部分。进程管理器的调度者按照特定的调度策略将任务映射到资源管理器中以使所有任务都满足需求。

2) 缓冲区管理。主机系统的有限内存带宽对多媒体应用带来很多的限制，这些应用需要大量数据的有效移动，而这却不能通过传统的数据复制来实现。类似的，目前操作系统中用于扩大内存的有效的页面技术和磁盘交换技术也不适用，因为多媒体数据不能放在真正的主存中以满足时间的需求。

3) 传输率和流量控制。因为多媒体数据以某个协商的传输率发送和使用网络资源，网络多媒体系统通信协议必须包括基于传输率的流量控制。因为现有的网络操作系统并不支持实时的视频会议信息，因此要提供在局域网和广域网上的连接方案。

下面给出在实际系统的研制过程中遇到的问题和相应的解决方案：

(a) 路由和桥接功能。路由器和网桥不是用来处理实时信息的，为了克服这个瓶颈，视频网关可用作视频路由器，路由器和网桥不处理视频信息。

(b) 网络协议。在局域网传输的环境下，为了建立高质量的对话，必须保证会议视频数据的可靠传输，因此需要开发在 TCP/IP 和 UDP/IP 上的传输层协议用来控制实时的音频和视频信息流量，利用这个协议能可靠地传输控制信息，音频信号可以几乎是即时的传输，视频和数据信号在保证音频传输的前提下，做到最好的传输，从系统的角度来说，呼叫的建立和拆除运行于 TCP/IP，会话的数据传送运行于 UDP/IP 上。

(c) 带宽的拨出和分配。视频会议需要固定的带宽来传送高质量的音频和视频信息，

但是标准的网络（如以太网）不提供这种功能，因此必须提供这种带宽的需要。每个局域网上传输的 384Kbit/s 的会话需占用的带宽（ $2 \times 384 + 150$ ）KB 左右的包头，加起来约为 900KB，人约占 10MB 以太网的 9% 的带宽。视频网关控制网上的会话数，任何时候视频会议数据在网上传输的带宽不超过 10MB 以太网的 40%，因此不超过 4 对 384Kbit/s 同时的会话可在以太网上建立。会话的速率适当的降低，则同时建立的会话数也可以适当地增加，即也可同时支持 12 对 128Kbit/s 的会话。当网络负载过重，超过 70% 的平均负载，应该考虑交换式的集线器，以增加局域网的吞吐量。

（d）时钟同步。视频会议系统端站之间时钟是同步的，而有些网络是异步的，如局域网，因此需要有一个设备来完成。

4）差错控制。会议电视系统需要实实在在的可靠性，所以它的组成需要端对端的差错检测和差错纠正机制。

在差错控制方面必须做到：

- 差错检测机制应该嵌入在应用程序中。
- 时序错误的差错检测机制必须嵌入在系统层和网络层中。

差错检测机制应该嵌入在应用程序中。例如，在 H.263 压缩的 B 块中的 AC 参数的一些错误，不应该影响预测的编码数据（它们仅仅在一秒钟的一部分时间出现并且肉眼几乎看不出来）。但是如果帧界被破坏了，则差错不能被复原。这意味着在应用数据流中的结构信息必须进行保护，同时也意味着现有的差错检测机制，如校验和数据单元序列化等，应该进行扩展以便包容更多的信息。现有机制允许在较低级进行数据的破坏、丢失、重复和次序混乱的检测（例如，网络层的包），但是在应用层，差错检测却被忽略了。

时序错误的差错检测机制也必须嵌入在系统层和网络层中。这样的检测机制可以在发送方—接受方路径上为其他连接打开资源，并且也可能在路径上指示出拥挤点。

传统的差错校正策略不适用于视频会议系统，预防性的差错校正方案如前向差错校正（FEC）和信道优先级编码方案要更加适用。

在 FEC 中，发送方增加冗余信息以便接受方能够定位和校正位和位序列。为了使用 FEC，需要：

- 在发送方和接受方之间的连接的错误概率。
- 应用程序所需的可靠性。

FEC 方法其端对端的延时较低，在接受方和发送方建立控制连接和回放之前无需数据的排它性的缓存。它的缺点是仅仅只适用于包内的差错检测和校正，不适用于完整包，而且不能保证毁坏的或丢失的数据包的恢复，另外它的冗余信息也大大增加了吞吐量的需求。

### 9.1.5 视频会议系统的安全保密

随着计算机、网络、通信技术发展，视频会议系统会变成协同合作、信息交流越来越方便的手段。它可以广泛地用于政府、国防机构、工商企业等，会议内容常常会涉及到许多国家机密、军事情报、商业秘密及私人信息等。因此，安全保密问题是视频会议系统的一个重要问题。

#### 1. 视频会议系统安全保密结构原理

国际电信组织 ITU-T 在 1993 年专门制定了声视业务安全系统的建议 H.233，该建议规

定的安全保密机制适用于窄带声视业务，并遵守 H.221、H.230 和 H.242 建议。安全系统应用于两个终端或一个终端和一个多点控制器（MCU）之间的点对点连接，当然也可以扩展到多点环境中，具体满足：

（1）安全性与系统提供的其他保密业务无关，可以由其他机制如关于密钥鉴定和管理建议草案提供密钥，或者也可以手工输入。

（2）它适用 H.221 建议的视声成帧信号，其传输率是  $P \times 64\text{Kbit/s}$ ，这里的  $P$  是 1~30 的任何值，与 H.221 建议一致，帧结构本身没有加密。

（3）给所有用户的语音、图像和数据信号在同一密钥下一起进行加密，保证传输安全保密。

（4）系统所采用的加密方案是可扩充的模块化方法，它便于加密算法的更新。

（5）系统的安全保密机制能够工作在点对点的视频会议系统中，也可以工作在多点会议系统中，此时允许 MCU 进行解密。

图 9-14 给出了一个视频会议系统安全保密方法的框图。它的主要组成是加密模块和解密模块，加密模块接收视频会议系统终端用户数据进行加密，形成加密后的数据在网络上传输，解密模块接收加密数据进行解密得到用户数据。从图中我们可以看到，并没有给密钥的产生和管理作出较多的规定，开发者可以根据自己的需要和可能，为视频会议系统提供密钥并进行管理，密钥被加密器、解密器和数据同步使用。如果需要采用密钥，则必须通过控制信道发送一个加载新密钥的标志。数据加密由加密器进行控制，通过控制信道发送一个标志指示当前数据正处于加密状态；而解密器的相应标志指示在需要时对数据进行解密。

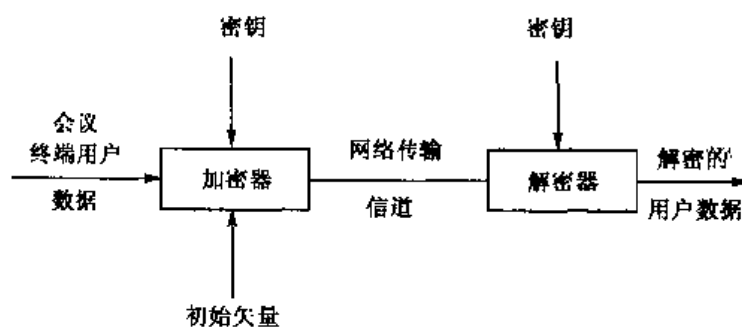


图 9-14 视频会议系统中安全保密结构框图

从应用角度来看，一个安全密码系统应包含如下功能：

- 秘密性（secrecy）。密文对非法接收者来说，不可被译。
- 可验证性（authenticity）。可验证信息来源的合法性，检验信息是否伪造，或以前信息的全发。
- 完整性（integrity）。可检验信息是否被更改，取代或删除。
- 不可否认性（nonrepudiation）。发送方对发送的信息不可否认。

早在中世纪，古罗马皇帝凯撒就开始使用密码，早期的加密方式主要是简单的代替和置换，借助的工具则是一些表格等，这是与当时的手工加密方式相适应的，这些密码的强度很低，可用统计方法等进行攻击。

此后，数论、近世代数等数学工具的发展和信息理论研究的深入，特别是计算工具的改

进，不断有新的加密思想和方法提出。现在，密码学的研究已取得了一系列较为系统的成果。密码系统按明文与密文的对应关系可分为序列密码和分组密码，其比较可参见表 9-7。

表 9-7 序列密码与分组密码的比较

	序 列 密 码	分 组 密 码
含义	一个明文符号转化成一个密文符号	一组明文符号转化成一组密文符号
优点	没有译码延迟，变换速度快	扩散能力强
	低错误扩散	在密文中插入信息破坏分组结构
缺点	低扩散能力	变换速度稍慢
	有可能拼凑出假消息	错误将影响同一分组

密码系统按加密密钥与解密密钥的对称性可分为对称密码系统（也称传统密码系统或私钥密码系统）、非对称密码系统（也称公钥密码系统）和混合密钥系统，其对比可参见表 9-8。

表 9-8 私钥、公钥及混合密码系统的比较

	私钥密码系统	公钥密码系统	混合密码系统
含义	加、解密密钥相同	加密密钥公开	用公钥加密会话密钥
安全性基础	第三方不能获得密钥	依赖于计算复杂的问题	用公钥加密私钥
优点	加、解密速度快	多用户时，能减少密钥量对安全性有计算复杂性的估计	兼有私钥和公钥的优点
缺点	密钥传输依赖于保密信道 多用户时密钥量大 安全性不易分析	加、解密速度较慢	
典型 体型	DES FEAL-N IDEA	RSA Rabin McEliece	MIX

视频会议系统涉及到加密算法范围极广，算法本身的说明并不包含在标准的建议中，但是加密算法的说明必须以某种方式给出，它必须包含下述细节：

- 初始矢量的长度和会话密钥。
  - 从初始化矢量生成起始变量等。
2. 视频会议系统中常用的加密算法

下面简单介绍一下视频会议系统中常用的两种加密算法：

(1) FEAL 加密算法。我们结合具体的实际应用对 FEAL 算法作一简介。视频会议系统的两个终端都根据当前的密钥和初始矢量生成码流，采用在 ISO8372 中定义的 OFB 模式中的 FEAL-8（8 round FEAL with 64-bit key）。在加密器方码流通过采用模 2 加和要加密的位进行组合，而在解密器方对同样的码流采用模 2 加恢复原来的用户信息。起动矢量（SV）和初始化矢量（IV）相等。IV 在每个多帧的起始处加载。加密算法的 64 位的输出中，MSB 方的第一个 8 位用来和声视信号块中的 8 位进行模 2 加；码字块的第一个位和信号块的第一个位进行模 2 加，并且结果位首先通过信道传送，码字块的第二个位和信号块的第二个位进行模 2 加，并且结果位第二个通过信道传送，依此类推。如果所有的 8 位都发送完了，那么

下一个周期的码流就接着生成并运用于加密中。

(2) DES 加密算法。数据加密标准 DES (Data Encryption Standard) 是分组密码体系中最成功的作品，也是运用的最为广泛的一种加密算法。它将 64 位的输入块变成 64 位的输出块，输出块的每一位都必须和输入块的每一位和 64 位的密钥块（其中 8 位是奇偶校验位）联合确定。DES 模式 1 将采用 OFB-0 方法，DES 模式 2 留待将来研究。起动变量 (SV) 和初始化矢量 (IV) 相等，参数标志符设置为[00000000]。

视频会议系统中允许多种加密算法并存，为此，必须有办法识别它们。我们采用一个字节表示算法标志。目前标志的几个算法如表 9-9 所示。

表 9-9 算法标志符及其定义

MSB	LSB	Meaning
0	0	Not allocated, Reserved for future use
0	1	"FEAL"
0	10	DES Model 1
0	11	Reserved for "DES" Model 2
0	100	Reserved for "DES" Model 3
0	101	Reserved for ISO/IEC 9979 algorithm register.
		Registration number 000001 (B-CRYPT)
Other values		Not allocated, reserved for future use

算法标志的默认值为[00000000]，通信设备至少应能解密所指定的算法中的一个；如果具有解密几种算法的能力，那么需要系统的操作员来指定选用哪一种加密算法。

在现有的视频会议系统密码体系下，加密和解密所用的密钥都是相同的。因而，密钥的管理就显得异常重要，这是因为密钥在使用了一定的时间以后就必须进行更改，否则就很难保证不失密，因为视频会议系统涉及的人员很多，而且加密方在每次启用新的密钥时，都要通过某种秘密的渠道把密钥传送给解密方。在传送过程中，密钥容易泄露。

可以用非对称密钥体制也称为公开密钥体制解决上述难题。系统工作流程如下：每个通信实体  $x$  都有一个私人密钥  $R_x$ （该密钥仅为实体个人所知）和一个公共密钥  $U_x$ （公共密钥为大家所知）。系统具有这样一个性质即一旦消息采用用户的公共密钥加密以后就只能用其私人密钥才能解开；反之亦然（但对于大多数的系统往往只是单向的动作）。所以如果发送方想向接收方发送一条消息的话，那么发送方首先查找接收方的公共密钥，然后计算密文  $C=E(M, U_b)$  并将密文发送给接收方。接收方使用他的私人密钥  $R_b$  通过计算  $M=D(C, R_b)$  而恢复出明文来，反之如果有人不知道密钥  $R_b$  而想恢复出明文是不可能的。如果消息  $M$  就是我们视频会议系统使用的会话密钥  $K$  的话，那么它就解决了密钥管理的问题。

事实证明采用公开密钥体制来进行视频会议系统中的会话密钥的管理是一种行之有效的方法。它解决了现有公共密钥保密系统的有效性不够高，保密性不够强的缺点。

访问控制的任务是：防止非法用户进入系统及合法的用户对系统资源的非法使用。可以采取两种措施：身份鉴别和使用授权。在过去，鉴别差不多总是和口令系统是同义词，但在今天，鉴别要做的事更多。例如，在一个视频会议系统中，一个多点控制器面临的是多个会场，多个会议，这时多点控制器就必须能够鉴别出用户是否是一个合法的用户。进一步，鉴

别系统还应提供单次登录的功能，这样用户就不用重复输入口令。虽然鉴别提供了身份的验证，但它并没有描述一个实体进程的优先权。例如，当你在加入到会议中来之前被鉴别并不意味着你有权去作任何事。后者的功能称之为授权。

对视频会议电视系统可以设立四级访问权限：超级、优先、一般及作废。

(1) 超级。这是为视频会议系统的管理员而设立的，他可以进行系统范围的安全控制和资源使用情况的审计。

(2) 优先。可以对系统的任何资源进行访问，可以申请作主席，可以申请数据令牌等。

(3) 一般。访问操作受到一定的限制，根据需要才可以让他申请主席、数据令牌。

(4) 作废。系统拒绝这一类的用户进行访问。

可以在每次会议预约的时候进行权限的分配，也可以在会议期间向系统管理员申请。

## 9.2 多媒体交互式电视技术

### 9.2.1 概述

#### 1. 交互式电视技术发展概况

有人说前几年世界的热点课题是多媒体计算机技术，近几年的热点课题是交互式电视技术（ITV-Interactive Television），这是因为交互式电视技术有较好的经济效益、社会效益和广阔的应用前景。系统用户可以坐在家里的电视机前，通过遥控器和菜单，在可能的条件下选择所喜欢的电影、电视和新闻。它还可以提供交互式电视教育、电视采购、视频游戏以及方便的电视、电话和数据信息服务。

交互式电视技术有较好的发展环境，自 1993 年 9 月美国宣布“信息高速公路”计划后，全球掀起信息高速公路热潮，投资巨款建设国家信息基础设施（NII），美国 NII 分三期：近期（1994-1997）、中期及远期（2000 年以后），投资 1500~40000 亿美元。我国也在积极、慎重地开展这方面的工作（CNII 计划），即在我国人力、财力、物力允许的条件下，为建立中国信息基础设施做技术和物质准备：建设“八横八纵”干线网计划正在实施，22 条光缆有 7 条始于北京；广州、上海，北京首先在国内引入高速率宽频带交换机 ATM；在近两年来我国出现了有线电视联网热潮，大到北京、上海大城市，小到县城、乡镇企业。目前我国已建有线电视台 600 多座，有线电视用户约 2000 万左右。

所谓交互式电视，是一种受观众控制的电视，在节目间和节目内观众能够作出选择决定，这是一种非对称双向通信模式的新型电视业务。节目间（interprogram）交互式电视可称为点播电视 VOD（Video On Demand），它又可分为真点播电视 TVOD（True Video On Demand）和准点播电视 NVOD（Near Video On Demand）。真点播电视是每个用户各自占用一套节目，即和电视交互的每个人都要及时响应，对装设在信息中心和电视台视频盘和视频带上的节目可以随意控制，提供上述所有服务及设备的费用是十分昂贵的。真点播电视的替代物是准点播电视，NVOD 是每隔一定时间（如 10min）从头播放一套节目，用户观看电视节目时，交换机将终端与最近将要从头开播的频道连通，用户等待时间不会超过时间间隔。NVOD 实现起来要便宜得多，它的延迟对用户的许多应用是无足轻重的，在等待时间还可向用户播放存储资料（像在剧院中看到的一样），或食品及其他产品广告，或提供音乐视频插曲，这样

使服务提供者能用广告利润补偿用户账单。节目内 (intrapogron) 交互式电视也称全交互型电视, 它能够将用户的请求应答即时传送给用户, 传送给用户的信息内容包括视频、图像、语音、文字和数据, 它可以用在教育 ITV、购物 ITV、交互式游戏 ITV, 但是实现交互式电影需要从一个场景切换到另一个场景的能力, 花费太大, 一般很少使用。

交互电视的最初应用将采用非对称通信模式, 即单向下行宽带传输, 应该指出的是在这些应用中, 导航和内容的无缝集成是重要的。典型的应用有:

(1) 电影点播服务 (MOD)。提供家庭 VCR 播放功能, 用户能够进行选择/取消、开始、停止、暂停 (冻结画面)、快进、倒退、向前/后查看, 设置或重置存储标志、显示计数器、跳到别的场景等。预看和交互浏览也是典型功能。其次, 应用还需要向用户提供诸如账单等数据。

(2) 远程购物 (TeleShopping)。商家利用多种媒体诸如视频、文本、带音频的活动视频、音频或图片 (静止或动画) 等技术制作虚拟商店, 用户在该店中浏览商品目录, 选择订购商品和所需的服务。

(3) 游戏 (Games)。与传统的游戏不同, 用户不是扮演指定的角色, 而是从菜单中选择角色。一旦选定一个角色, 游戏程序被装入到用户的 STB (机顶盒) 中或服务消费者的“游戏机”中。游戏过程中用户的输入将改变游戏的状态, 并被以图画/视频流方式呈现给用户, 游戏继续下去, 直到游戏自行结束, 或者由用户决定结束。

(4) 卡拉 OK 服务 (Karaoke on Demand)。用户位于不同的地方, 从服务提供者给出的目录中选择歌曲, 用户可以选择或改变音调和节奏。当用户对着麦克风唱歌时, 器乐声和唱声在用户所在的地方放出来。

(5) 新闻点播服务 (News on Demand)。用户可以交互地选择新闻项目、表现的类别和材料 (如正文, 或连带活动视频)。

(6) TV 列表 (TVListing)。用户可以看到一个个节目以及它们的安排。用户可以选择安排的节目, 以便在进一步查询时得到其他信息 (如状况、演员、制作时间等)。

(7) 家庭银行服务 (Home Banking)。应用电子访问由银行提供的服务, 包括查看账目余额、向第三方付款、申请借贷以及浏览银行的服务情况。

(8) Internet 访问 (Internet Access)。主要用户通过 STB 使用 Internet 提供的服务。可以有多种配置: 如 STB 作为通信设备, 一端连接到 PC 机上, 另一端与宽带网连接, PC 通过 STB 访问 Internet 信息、下载文件、发送邮件等; 又如 STB 只作为一台远程终端, 电视作为它的显示设备; 再如 STB 作为一台 PC 机。

美国电话公司 (TELCOS)、有线电视 (CATV) 公司正在致力于研究和开发 ITV 技术, 它们既是合作者又是竞争对手。按惯例电话公司支持交互性的实现, 它拥有非对称数据用户网络技术 ADSL (Asymmetric Data Subscriber Loop), 利用现有的双绞线向大多数用户传递 2~6MHz 带宽的双向数据, 向家庭和商业企业传送图像、电视和数据, TELCO 在美国已拥有 9400 万个家庭用户。有线电视公司拥有大量单向带宽通路, 为 5800 万家庭提供灵活大容量的光纤网络用以传送各种图像、视频、音频及计算机数据, 并具有播出多种电视节目的经验。

早在 1986 年, 南贝尔 (Bell South) 公司就开始进行了 VOD 的有关实验。1993 年, 大西洋贝尔 (Bell Atlantic) 公司在美国政府资助下, 开始在电话线上进行了电视视频拨号音

的实验,并试验将 MPEG-I 的压缩数据节目传送到 2000 个用户的机顶盒,通过机顶盒解码给用户。在这一年, Nynex 公司也开始向曼哈顿居民提供他们的第一套试验性的点播电视系统。1994 年 12 月 14 日,在美国召开了 Time Warner 全业务网(Full Service Network)的新闻发布会,称这个日子为点播电视的誕生日。从此,点播电视系统进入蓬勃发展的时期。IBM 公司在 1995 年 7 月宣布建立交互式电视应用开发实验室,选用 Microware 公司的数字音频/视频交互式解码系统(Digital Audio-Visual Interactive Decoder—DAVID)作为点播电视的标准和开发平台。近些年来,美国的许多大计算机公司都已经涉足点播电视系统这一领域。许多公司设计出自己的视频服务器推向市场,机顶盒的大规模设计开发也进入实质阶段,很多的芯片设计厂商设计了专为机顶盒的芯片;软件厂商也参与其中,Microsoft 公司收购美国最大的机顶盒生产厂商,同时大力推动其 Windows CE 作为这类设备的操作系统,今年 Microsoft 的总裁 Bill Gate 在深圳推出的“维娜斯”计划也是基于这个考虑。目前全世界已有 17 个国家正在开发或者提供点播电视的实验和服务。在亚洲,韩国的 KT 公司已经推出了其点播电视业务,通过电话线向家庭提供点播电视服务,并打算将这个系统的规模拓展到上千个家庭并开始商业化。日本 NEC 等公司开发了大型的视频服务器,并在东京地区建设了大规模的点播电视的实验平台。在中国,1996 年,海南三亚信息工业公司(SII)与美国微软公司合作,在上海地区进行了交互式电视(MITV)的实验,参加的单位有上海信息工业公司、上海教育电视台、上海广播电视台以及上海电信局,于 1996 年投入市场实验。1997 年,广东在其全省建设了综合业务宽带网络,并在上面开展点播电视业务。1999 年初,四川成都开始实验基于机顶盒接收的有线电视业务,这个实验无疑对今后的点播电视系统打下良好的基础。与此同时,各种小型的点播电视系统在中国也飞速发展。基于小型服务器和局域网环境的公司用培训点播电视系统,旅馆用娱乐点播电视系统,学校用教学点播电视系统都已经出现,尽管这些系统目前尚有不少的缺陷,但是它毕竟标志着点播电视系统已经向我们的日常生活走来。

## 2. 交互式电视技术和分布式多媒体系统

交互式电视和分布式多媒体技术有很多相同之处。一个局域网(LAN)分布式多媒体系统及两个局域网多媒体系统通过桥接器和路由器(Brouter)相互连接的结构,它们的终端可能是工作站,都具有交互性,能够显示完美的带有伴音的连续运动的视频图像,看上去和电视机一样,能够显示让人喜爱的影片,它可以显示训练信息,个别指导,研究数据和其他有关的商业信息。例如可以是一个位于一家汽车商店的系统,一部分工作站放在展示大厅,配合顾客选购汽车,另一些工作站放在服务部门用于机械师修理汽车,诊断汽车的毛病,给出正确修复的方案。同时,销售人员还可以用这套系统进行调查,与竞争对手比较某种汽车的优越性。它可以是一个小型只包括自己视频服务器的局域网,它也可以扩充到由一些汽车销售商的局域网相连,以便共享信息。还可以给出一种利用有线电视电缆(CATVCoax)连接有视频服务器的交互式电视系统。每一台电视机通过机顶盒和网络,用交互式方法使用户选择想看的电视节目。如果一个 90 分钟的节目,采用时间重叠每 6 分钟重播一次的方式,那么整个节目需要分 15 部分播出,用户从点播到收看电视节目最大的延迟时间是 6 分钟。

随着多芯光纤的出现,它将在光缆干线上提供更高频率的带宽,这样的系统可以很方便地将交互式电视和多媒体网络系统融合在一起。提高带宽的有效方法是用户和服务器分成组



或小的单元，结点分组的秘诀是中心服务器和用户之间的信息只在某个特定的组内流动，不允许流到其他小组，除非信息被特定的组外其他用户索求。组到组之间的通信需要宽带光纤和网间连接器（交换器）提供通道。

交互式电视和多媒体一个重要的区别是信息传输采用了不对称模式，交互式电视系统和人类间直接通信的信息系统一样，数据的传送量和接收量有大量的差异，人类的眼脑和耳脑结合可以迅速地接收非常多的信息，而操作键盘和定位设备的手就要慢好几个数量级。使用遥控器的交互式电视系统，目前需要以每秒兆位的速度传送家用质量的电视，而从遥控器到机顶盒只是每分钟几位的数量级。

这种不对称在多媒体通信、分布式多媒体系统以及计算机网络系统中，相对来说是一个新概念，上述系统两个方向的传输路径具有相同的传输率，数据交换是处于计算机之间而不是人和机器。交互式电视系统通信对象是人和机器，所以它把传输通路分成节目通路和返回通路。节目通路也称下行通路是流向用户，是把视频信息传送到用户；现在有很多技术可以传送视频，这些技术共同的特点是“高频宽带”，例如有线电视的交互式电视系统把 50M~1000MHz 分配给下行通路。返回通路也称上行通路，通常是用户把他的需求通过单键遥控器、机顶盒进行：节目选择、登记的响应（电子会议）、响应教育节目的测试问题；从电视购物目录中浏览和预订等。它的数据频率很低，有线电视交互式电视系统把 5~42MHz 分配给返回通路。

### 3. 交互式电视系统一个示例

一个 ITV 用户正坐在他家的交互式电视机旁，正考虑今天想看什么节目，当他拿起非常简单的单键遥控器时，一系列小的矩形菜单出现在电视机屏幕的顶部，如图 9-15 所示。

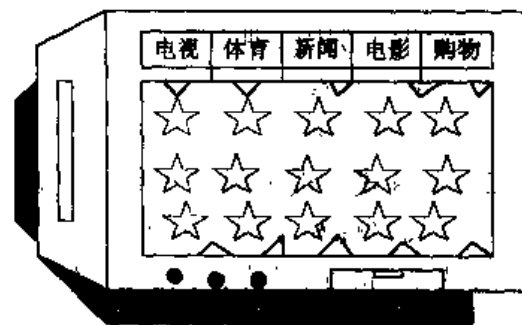


图 9-15 交互式电视菜单系统

他想看一部电影，于是就把单键遥控器（具有激光指示器）指向菜单的“电影”框，于是菜单向下扩展，把电影分成 5 组：浪漫传奇片、动作惊险片、经典著作片、科学幻想片及滑稽喜剧片。用户选择了“动作惊险片”，于是在一系列电影片名中有一部史泰龙的“攀岩者”，这正是用户想看的，他选完电影后感到饿了想点购食物，五种诱人的比萨饼出现在屏幕上：胡椒的、蔬菜

的、西西里式、还有乳酪的以及混合式的比萨饼。用户指示出选择所需品种，屏幕通知用户在按键选择后 30 分钟内比萨饼将送到，否则免费。用户不必告诉比萨饼需要送到哪儿，因为系统知道他是谁，住在什么地方，接下来是选择饮料，选完后，完整的食物订单显示在屏幕上以便确认，同时还出现一些电影的节选镜头，使用户确认是否是他真想要看的电影，给予一个改变主意的机会。当用户被告之价格信息后，电影真正开始了。

从开始选择到播放电影大概花了两分钟时间，整个过程，用户的需要得到了满足。电影在播放，用户从中得到消遣。电影开始不到半小时，门铃响了，是送比萨饼的人，用户从电视机旁起身，但不需拿什么钱，ITV 系统已经把钱付给了饼店，用户打开门，付给送货人小费，拿了比萨饼到厨房取些餐具，然后回来接着观看电影。

然而，用户错过了电影中三分钟最好的情节，他拿起遥控器，大屏幕上立即出现十五个

小屏幕，每一个都放着同一部电影，但是开始时间不同，用户选择了在他刚刚看过后面的一个（延迟了三分钟的同一部电影），因为他记起小屏幕放的正是比萨饼送来之间的情节，用户从他刚才被打断的地方继续观看。

我们的主人翁把一切安排好后开始用餐，电影又放了 12 分钟后，出现了一个令任何进食者会作呕的画面，他又拿起遥控器，大屏幕上又出现了那些小屏幕，用户可以比快进盒式录像机更容易绕过一些恐怖情节，实际上用户可以随意访问任何一个镜头。

## 9.2.2 交互式电视系统的功能和结构

### 1. 交互式电视系统的功能

交互电视系统要适应广泛的应用领域，其许多功能分散在各子系统内。这些功能有共性功能和特殊功能，未来应用还将定义和增加新的功能。

从系统的操作、完整性和开发方面看，交互电视系统核心功能应包括：位传输、会话、访问控制、导航与节目选择、应用启动、媒体同步链、应用控制、表现控制、使用数据（用户使用系统信息）和用户概况等。

从交互电视应用与服务角度上看，交互电视系统应具有如下公共功能：

- 导航与交互功能。导航是指在菜单等导航信息引导下提供内容的查找、选择。交互功能包括交互演播控制（如仿真 VCR 控制回放）和多媒体交互表现。
- 服务与内容管理。交互电视是大众家庭消费应用，除提供应用服务外还需要考虑用户使用服务的记账和用户的付费等；为此系统应标识、定位和跟踪内容材料的管理。此外系统应具有能指明错误并恢复、插播商业信息、收集和统计使用信息、多重内容访问及进行监控操作和维护等能力。
- 知识产权（IPR）和安全交互电视应用应能提供安全的进程、机制、协议、通用和专用的算法及对象（如密钥与随机数、安全盒、安全卡等），进而保证应用与系统的安全。交互电视系统还要保护内容的版权和任何合法者使用系统的权力。

除此之外，复杂的交互电视系统还应解决分系统之间信息的互通性，应用在异种平台的可移植性以及交互信息在传递过程中的延迟等应用问题。

交互式电视系统最常用的是节目间的交互，又称点播电视（VOD）。按照点播电视系统服务的规模，将点播电视系统划分为三类：

（1）小型点播电视系统。这种服务器的应用领域是 50 点以内的小型网络。利用小型点播电视系统可以构造多媒体教室，在多媒体教室内，学生可以点播视频的教学节目，观看辅助教学的参考录像等。小型的点播电视系统还可以用来在公司内建构公司培训系统。公司职员通过公司内部的内网（intranet）选择自己的空闲时间来收看培训的视频节目。这样的点播电视系统还可以作视频信息查询系统，在车站、银行、邮电旅游等服务行业，小型的点播电视系统可以取代人工问讯机构，提供直观的视频信息服务。

（2）中型点播电视系统。这种系统的用户数目在 50~1000 之间。这种系统特别适合于作饭店、歌厅等处所的娱乐点播系统。用户可以在饭店的房间内点播收看电影、电视和新闻节目，也可以通过这个系统来点播卡拉 OK。

（3）大型点播电视系统。大型点播电视系统可服务的用户数目至少在 1000 人以上，大型点播电视系统是未来城市有线电视的发展方向，未来的有线电视在兼容现有的广播模式电

视节目的同时，将支持用户对特定节目的点播需要。这种有线电视系统是现有广播电视、视频节目出租、信息查询等多种功能的综合体。通过这个网络，用户甚至可以浏览商店的货架，进行网上购物；点播交互式游戏，连接成千上万用户参与大规模的游戏等。

2. 交互式电视系统的体系结构

为了说明 VOD 系统的工作原理和结构的组成部分，我们先给出通用点播电视系统主框图和通用系统图。如图 9-16 通用 VOD 系统主要的组成部分有：

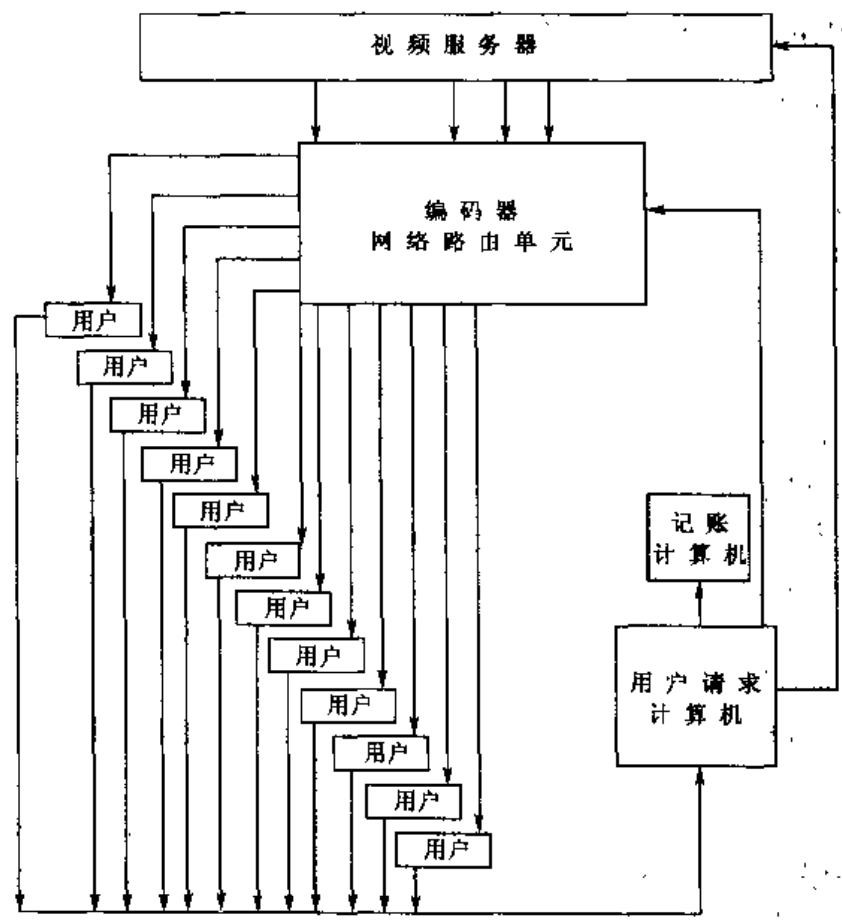


图 9-16 通用点播电视系统方框图

- 视频服务器。
- 编码器/网络路由单元。
- 用户请求计算机和记账计算机。
- 位于用户处的机顶盒。

较详细的通用 VOD 系统结构如图 9-17 所示，它把系统分成两部分：一部分是节目分配的设备，它包括信号融合打包器、接收器、用户选择计算机、记账和登记计算机、服务器、编码器、混合器、调制器以及传送器；另一部分是用户部分，它包括信号解包器、解调器、解压缩、宽带视频/音频电视接收机，用户接口及节目选择接口。

(1) 视频服务器。VOD 系统为了进行点播电视服务需要一个大容量存储器，并具有高速传输能力的快速检索系统，这就是视频服务器，它的主要功能是：

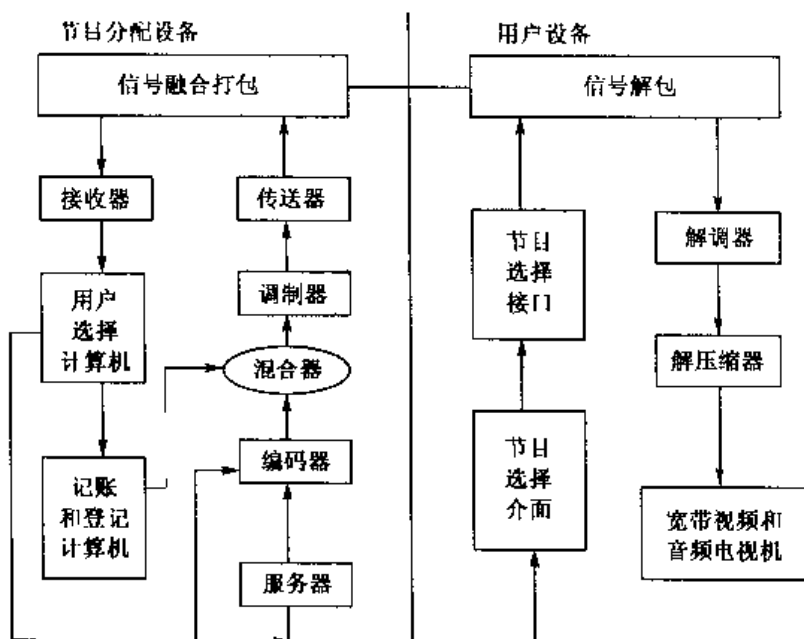


图 9-17 通用点播电视系统节目分配设备和用户终端方框图

1) 大容量视频存储。典型的视频存储器可以有 20GB 的存储空间 (20 部 90 分钟的电影), 采用多重单元互连可以提供几倍 20 部影片。视频服务器选用存储体有磁盘、磁带和 RAM, 为了提高性能价格比, 较多的采用温彻斯特硬盘, 现在单盘可以以 2Mbit/s 速度产生 8 路视频数据流。为了提高数据传输率、I/O 响应时间、可靠性、存储容量及降低费用, 大量采用廉价磁盘冗余阵列 RAID (Redundant Array of Inexpensive Disks) 技术, 制作视频阵列盘。要满足对不太流行视频节目的需要, 必须有二级存储器, 通常采用智能化磁带 (robotic tape libraries)。由于 RAM 费用太高只少量用在 Cache 缓存中。

2) 节目检索和服务。视频服务器接收所有用户的全部的信号对服务器进行控制, 服务器控制处理能力要根据应用的不同进行设计, 例如, 交互较少的点播影片只需要较少的控制处理能力; 而对交互式较多的交互式学习, 交互式购物及交互式视频游戏, 就需要高性能的计算平台。

3) 快速的传输通道。视频服务器有一个高速、宽带的下行通道与编码路由器相连, 把服务数据(视频、音频及嵌入控制信号的数据)传送给各个用户。Silicon graphics 宣称它的 Challenger XL 系列(用在 Time-Warner 的 Full Service Network 系统中), 通过总线每秒可以传送大约 1GB 数据。同时, 视频服务器接收来自用户请求计算机来的各种反向通路信号, 每个用户要求的带宽较低(16Kbit/s), 当然数据量和速率也取决于用户的多少。

(2) 编码器/网络路由单元。编码器/网络路由单元是按用户的需要, 将节目和服务数据编码经过路由选定分配到用户。TELCOVOD 系统按用户的需要将节目和服务数据分配到不同的双绞线进行空间划分, 每根双绞线通过这种方式携带一个节目, 送到用户家中。其他的 VOD 系统, 有的采用时分和频分复合方法, 将节目送到目的地, 还需要用户机顶盒中信号分离器和解码器才能最终把需要的节目惟一的送到用户的电视机上。

节日线路确定在分组交换网中得以最佳实现,这是路由的自动实现,每一个分组都含有目的地,信息源及节日地址等信号,SONET、FDDI 和 ATM 都是很有生命力的实现方法,

但对于视频分配，ATM 可能是较好的方法。

异步传输模式 ATM 是由许多 53 个字节的定长包组成，可以异步地插入数字信号和它们的路由信息。虽然每个单元与其他单元之间是异步的（没有联系），但特定用户或服务所使用的单元保持在一个固定的顺序和时间域中。ATM 的单元有 53 个字节长，其中 5 个字节是含有地址的首部和一个固定 48 个字节的信元域。

ATM 路由器提供 ATM 信元数据单元和路由选定，由 ATM 单元首部中的地址控制，通过自选定的程序段与视频服务器进行传送。ATM 路由器提供必要的电气接口，从视频服务器接收数据并分配到用户网络中去。ATM 由交换器通过外部的 ATM 到 OS-S 多路转换器直接连接到电话交换载体进行远距离服务。

ATM 调换器/路由器将需要：

- 53 字节单元、格式符合 CCITT 和 ANSI 规定。
- 物理接口符合 DS-1、DS-3 及 STS-3C。
- 视频服务器接口兼容性。
- 可扩展和模块化结构。
- 容错能力。
- 基于应用标准，开放系统设计（包括交换控制器和操作系统软件）。

(3) 用户请求和记账计算机。在 VOD 系统中每台用户请求计算机可以支持 6000 个用户，多个用户请求计算机可以将几倍的 6000 用户增加到 VOD 系统中，每台计算机需要足够的物理空间支持 6000 个 19.2Kbit/s 的逻辑端口。（一块印制电路板有 200 个端口，共有 30 块通信板），并有能力允许同时进行分组通信。用户请求计算机将通过机顶盒提供立即或接近立即响应，就像在多用户分时系统中一样，当不能提供立即响应时，需要一些提问，叙述和其他短节目插入，这仍会使用户感觉到能够得到立即响应。当选中一项到接收到新的菜单或节目，不会超过 6 秒种的延迟。

信息传输的延迟是由于过多的用户同时访问系统，而大部分延迟是由于用户请求计算机正处理大量事件。例如，一个有 4500 个用户正观看的视频精彩节目在晚上 8 点结束，节目结束后，有 4300 个用户在 1 分钟之内选择新的节目。这样一秒种之内将有 72 个复杂事件等待处理，用户请求计算机将能很好地满足这些需要。

用户请求计算机和记账计算机还解决服务管理问题，它包括服务类型，更新修改服务及各种服务的收费问题。

(4) 电视机机顶盒。电视机机顶盒 (TV Set Top Box) 是 VOD 系统用户购置的用于节目选择与 VOD 系统交互通信的用户室内设备 CPE (Customer Premises Equipment)。它应该具有以下功能（详细结构见第五节）：

- 指示用户室内设备，网络传输和节目资源的状态，在用户的电视屏幕上显示服务公司及信息提供者所给出的信息和菜单。
- 把用户的选择传送给服务中心或信息提供者。
- 提供用户基本的控制功能，如电源的开/关，选择 VOD 或标准电视操作。
- 控制用户的电视或盒式录像机。
- 交互式购物。
- 交互式教学。

- 与家用计算机接口。
- 数据传输。
- 公用设备监测。
- 传输性能的遥测反馈。

### 9.2.3 视频服务器的结构和设计

网络多媒体的发展方兴未艾得益于两方面：多媒体促进了计算机人性化，网络促进计算机普及化。传统计算机是面向计算而设计的，其功能简单且操作复杂，因此制约了人们对计算机的应用，而多媒体则改变这一局面。狭义上多媒体指视频、音频等媒体技术；广义上多媒体指人机交互的各种手段和方法。在计算机硬件水平按照摩尔定律飞速发展高潮时期，视觉、听觉、逻辑、感性，多媒体和人机交互技术在降低计算机使用难度的同时，还提高了信息的表现能力，促进了计算机向人性化方向发展。网络技术的发展则使大量信息涌入人们的生活，人们对信息的渴望更促进网络的进一步普及和发展。Internet 的发展极大地提高了人们获得信息的能力，世界正在变成一个“地球村”，人与人之间的交互已经非常简单。在这样的背景下，网络多媒体的发展成为一个必然结果。

Internet 和 World Wide Web 的快速发展使得人们可以方便快捷地访问到存放在不同物理位置的服务器上的各种信息，成为一次新的信息技术革命。目前，存放在服务器上的信息主要为文本和静态图像。随着网络带宽和计算机处理能力的提升，人们的注意力转向连续媒体，越来越多的人希望能够通过网络欣赏热门的音乐、观看最新的影视节目。根据一些机构的统计，在 1998 年的前 9 个月，网络服务器上的视频和音频内容增加了 3 倍。预计到 2003 年，存储在网络服务器上的 50% 以上的内容将是连续媒体信息。互联网正从一个静态数据内容（文本、图像）占主导地位的阶段快速地发展成为一个包含静态媒体内容和连续媒体内容的综合的信息资源库。

#### 1. VOD 和视频服务器系统

VOD 系统中最关键、技术含量最高的是视频服务器。从计算机的技术角度讲，视频服务器是一台或一组提供媒体存储和媒体传输的计算机。其基本功能是存储并传输媒体节目。之所以将视频服务器的概念与普通计算机相区分，是因为视频服务器在存储和传输媒体节目的能力上显著强于普通计算机，而媒体节目的存储和传输是 VOD 系统功能实现的核心，VOD 系统的服务能力几乎完全由视频服务器服务能力决定，因此学术界和产业界对 VOD 的研发重点都集中在视频服务器上。

视频服务器必须具有巨大的存储能力。普通 VCD 影片（MPEG-1）数据大约为 1GB 左右，而图像质量更好的 DVD（MPEG-2）影片数据量则为 4GB 左右。即使仅存储 100 部 VCD 影片，服务器的存储容量也要达到 100GB，这对一般计算机而言难以实现。单台的视频服务一般采用大型硬盘阵列，并辅以磁带机、光盘塔等其他的大容量存储设备。但即使是承载量为 100 张光盘的光盘塔，也仅为系统添加了 50 部影片，对于千户规模 VOD 系统而言，这种层次存储的食品服务无法满足要求。典型的解决办法是建立分布式机群结构的视频服务器，建立以单机视频服务器为基本单元的分布型层次存储模型。这也是目前视频服务器研发的重点问题。

假如视频服务器采用磁带机、光盘塔等大容量存储设备，还存在如何有效传输媒体数据

的问题。媒体数据的传输同文字、图像等数据传输相比有很大差异。首先数据量庞大，一张《人民日报》的文字不超过 1MB，1MB 也可以传输几十张精美的图片，但对视频而言，即使是低清晰度的 VCD，1MB 也仅能播放 10s 节目，用户在相同时间消耗下，对不同媒体的数据量需求差异巨大。其次媒体数据是实时性数据。Internet 的典型应用对实时性要求都不大，例如消耗几秒下载网页可供观看很长时间，这些数据没有明显的实时性要求，而媒体数据的数据量巨大，用户不能容忍先下载再收看的模式，而必须边下载边收看，即必须实现媒体的数据流化和流式媒体播放技术。对视频服务器而言，如果节目采用 1.5Mbit/s 的 VCD 影片，单台服务器同时支持数十上百路数据，这对存储设备和网络设备的 I/O 吞吐要求非常高，简单利用磁带机、光盘塔甚至一般硬盘都无法满足要求。为解决这些问题，必须在计算机硬件体系结构和软件算法两方面，对视频服务器进行专门的设计。

视频服务器提供给用户的点播服务分为不同类型，其差别主要体现在用户对节目的控制能力上。准点播电视（NVOD）将节目每隔一定的时间从头播放，例如两小时长的节目每隔 10 分钟播放一次，则共需 12 个单独频道，即生成 12 个媒体流，用户只能从 12 个频道中选择。这里存在等待延迟，等待延迟对某些类型的用户没有明显影响，因为延迟时间比较短，还可利用等待延迟播放广告补偿用户的帐单。与准点播电视不同，真点播电视（TVOD）每个用户独占一个“频道”，视频服务器必须实时响应用户命令。即使两个用户的节目要求仅差 1 分钟，系统也必须提供精确的服务。所以，真点播电视在技术和经济两方面同准点播电视都有很大区别。上述两种点播电视都属于节目间点播，用户只能点播节目并等待节目的开始，而不具有对节目的操作能力，例如暂停、快进、快退等（因为传统的录像机 VCR 都支持这些操作，所以它们又被称作 VCR 操作）。交互式点播电视（IVOD）支持这些 VCR 操作，因此其技术实现更加复杂。视频服务器在发展初期仅支持准点电视，因为网络传输带宽有限，视频流的数目有限，因此不同用户共享频道而节省网络资源；准点播电视管理相对简单，系统也实现容易。随着技术发展，视频服务器可对更多资源进行有效管理并能实现更复杂的调度和操作，实现真点播电视以及交互式点播电视成为目前视频服务器的基本功能。

不同视频服务器的规模也有显著差异。小型视频服务器支持 50 点以内的用户，多媒体教室是使用小型视频服务器的例子。中等规模网络的视频服务器支持的用户数目在 50~1000 之间，例如旅店娱乐点播电视系统需要中型视频服务器。大型视频服务器支撑城域网络。未来城市有线电视网将发展成为城域点播电视系统，在这种范围提供点播服务的视频服务器将是一个大型的网络体系。不同规模视频服务器的设计和实现差异很大：通用计算机通过改造就能完成小型视频服务器的功能；中型视频服务器是目前的研究热点，现已提出许多解决方案和算法；大型城域级别的视频服务器仅处在理论研究和探讨阶段。

视频服务器不仅可应用于 VOD 系统，其支持的图书、期刊、报纸、图片、声音、视频以及科学数据等数据的点播在远程教育、数字图书馆、新闻点播、电子采购等文化、商业、娱乐等各种领域，都具有很高的实用价值。从视频服务器的技术角度讲，它是连接网络提供视频存储和传输服务的设备，即沟通网络业务和多媒体业务的桥梁，因此，对视频服务器作一定改动后，也可在许多网络多媒体业务提供支持。例如改造视频服务器成为网络监控系统录像中心，多路监控视频通过网络传输至视频服务器并存储。改造视频服务器成为视频会议控制单元，会议参与者将各自视频音频数据提交会议控制单元并重新发送到每个会议参与者。因此，成熟的视频服务器技术将带动整个网络多媒体应用的发展。

## 2. 视频服务器的体系结构

从现有产品可以看出,不同视频服务器在设计思想、应用领域和实现上有不同着眼点,但视频服务器要解决的问题基本相同,即提高数据存储量、增加数据传输能力。为解决上述问题,研发机构提出不同的视频服务器体系结构。从技术发展角度看,在视频服务器发展的不同阶段人们对视频服务器的理解不同,厂商对其产品在经济和技术的综合理念不同,因此不同视频服务器的体系结构也受影响。

### (1) 视频服务器的类型:

1) 基于通用计算机的服务器。这类服务器采用 PC 等通用计算机作为主机,以硬盘为主要存储介质,面向小型网络用户,其性能一般可支持 50 点、MPEG-1 视频点播的要求。由于服务用户数目和服务质量有限,系统要解决的核心问题是网络传输和管理,以及视频数据存储和 I/O 吞吐效率等问题。以支持 50 点 MPEG-1 服务为例,50 部常规影片的存储要求是 50GB,而同时播放 50 个视频流对 I/O 接口的要求是 75Mbit/s,这种要求一般计算机很难达到。目前解决这一问题比较成熟的技术是基于 RAID 阵列盘和 SCSI 高速接口的存储体系。从系统软件上看,这种服务器的功能比较简单,一般仅作基本的流高速处理。这种服务器构成的 VOD 系统关键要解决视频流在网络上高速、有 QoS 保障的传输。因此,视频服务器在硬件上可采用高速网络体系结构,典型如 ATM 网络等;在软件上要设计实现支持流式数据传输的协议和协议族, IETF (Internet Expert Task Force) 针对这一需求提出了 IPv6、RTP (Real-time Transfer Protocol)、RSVP (Resource ReSer Vation Protocol) 等协议族。当然不论视频服务器结构如何,从 VOD 系统考虑,网络传输都是一个很关键的问题。Starlight 公司 StarWorks 的核心技术为其网络协议 MTP (Multimedia Transfer Protocol) 就是基于上述考虑。

2) 基于高级工作站的服务器。基于高级工作站为大中型计算机公司的设计思路。硬件技术的发展使新型微处理器的处理能力超过中型机的处理能力;而且随着军用市场的萎缩,大中型计算机公司还要为其产品寻找新市场,视频服务器成为理所当然的新应用。这种类型的视频服务器往往继承原有技术,在高性能计算机的基础上增加支持视频数据访问的有关硬件,并将系统进行一定程度的优化,使之达到一般视频服务器的功能。

以 SUN 公司的 MediaCenter UltraSPARC 为例,如图 9-18 所示,它的主机为 UltraSPARC 服务器,软件也采用原有的 Solaris 操作系统。为支持视频服务器功能,MediaCenter 硬件上采用基于交换式的两个 UltraSPARC 处理器结构,其处理器、系统内存及输入/输出均相互连接,存储采用 SCSI 接口;MediaCenter 的软件结构在 Solaris 基础上,在底层优化了网络与磁盘驱动,其核心部分还提供了视频节目与流性能的管理功能。

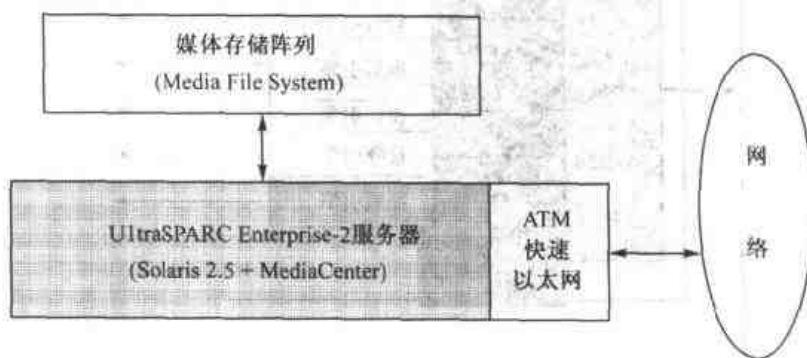


图 9-18 MediaCenter UltraSPARC



Concurrent Computer 公司的 MediaHawk 也是基于高级工作站的视频服务器。该公司生产实时计算机应用于导弹、飞行器的飞行模拟、数据获取等军事领域，实时的系统经过进一步的优化后非常适合交互式视频点播的需求。MediaHawk 优化的操作系统 PowerMAX OS 是专为实时应用设计的操作系统，它在内核级支持高度优先级别，因此降低了流高度中的用户响应时间；其磁盘存储管理系统优化了连续数据文件的存储访问，磁盘管理系统将磁头移动和磁盘旋转等待时间降低到很小水平；操作系统为 I/O 系统和存储系统提供了相连的通路，降低了数据从存储系统到网络接口的时间延迟。这些为支持实时系统的技术保证了在相同硬件条件下 MediaHawk 比其他视频服务器有更高的服务能力。

基于高级工作站的视频服务器设计思路非常简单，充分利用已有计算机结构，充分利用其硬件性能。基于高级工作站的视频服务器服务能力可以支持中等规模的网络需求，其缺点是价格昂贵。SGI、Oracle、DEC 等都生产这类服务器。

3) 基于专用硬件平台的服务器。基于专用硬件平台视频服务器和基于高级工作站视频服务器设计思路相似，其体系结构由生产厂商的生产基础决定。工作站厂商继承原有计算机体系结构，而硬件厂商则偏向于从以专用硬件单元组合的方式解决问题。视频服务器发展初期的核心问题是存储和 I/O 问题，硬件厂商生产专用的大容量存储设备和高速 I/O 设备构成基于专用硬件平台的视频服务器。典型的基于专用硬件平台的视频服务器是 FVC 的 V-CACHE，从体系结构上来说，V-CACHE 不是普通概念下的计算机，而是一个有独立结构的硬件单元；从功能上讲，V-CACHE 除视频服务外不能从事任何计算工作。

基于专用硬件平台的大型服务器一般面向有线电视领域，服务较大范围内的许多用户。这种类型的服务器价格较高，性能也很优异，但是其对硬件的依赖性很大，系统的专用性很强。以 Video-Active 公司的视频服务器 ADS (All Demand Server) 为例，如图 9-19 所示，其核心为数字视频引擎。数字视频引擎内的 RISC 处理器管理一组 SCSI 硬盘，连接一个高速的视频输送接口。这样的数字视频引擎就是一个单独的视频处理器单元。而主机的作用仅仅是作这个视频服务器单元阵列的管理计算机。由于阵列有可扩充性，ADS 的服务能力很强。

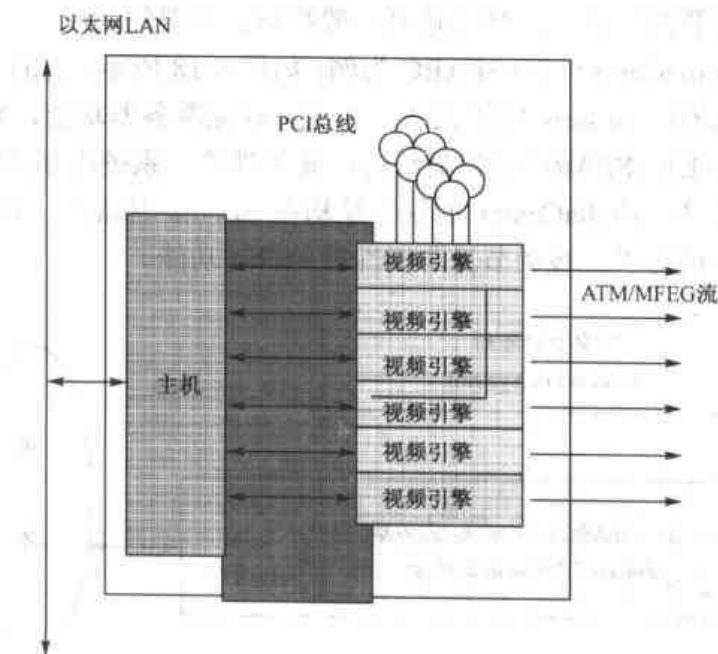


图 9-19 ADS 体系结构

大规模并行计算结构是通过高速互联网络，将每个自带存储设备 I/O、网络输出接口、内存等的处理机连接起来，组成高性能并行计算机，满足视频服务器的设计需求，基本模块是构成视频服务器基本单元。nCube MediaCube 以及 SGI Original 系列视频服务器采用的是这种体系结构。

MediaCube 视频服务器在可扩展性方面设计得很有特色。MediaCube 的每个模块设计都对 I/O 资源和处理能力进行了仔细配置，以适应视频点播应用的需求。通过其 HyperCube 专利互联网络连接这些基本模块，在系统扩展时保证各种资源的平衡。每个 MediaCube 基本功能模块由两个节点构成，每个节点是一个独立的实体，它们拥有各自的内存，独立地运行各自的 Transit 操作系统和 Oracle Video Server 软件。因此，每个节点在执行任务时不会影响系统中其它任何一个节点。MediaCube 基本模块采用 hyperCUBE 拓扑结构互联，在增加模块数量的同时，增加各模块之间的通信带宽和通信处理能力。

基于大规模并行计算机结构的视频服务器系统具有良好的系统可扩展性，通过互联不同数目的基本模块，组成不同服务能力的视频服务器系统，可以满足从小旅店到城域级电视点播不同规模的应用需求。由于采用大规模并行计算结构，使视频服务器造价极其昂贵，限制了它的应用范围。

基于高级工作站和专用硬件平台的视频服务器面向高端用户，随着技术的不断提高，以这类服务器作为今后城域级视频服务点播网络的节点是一个必然的趋势，但目前在面向中小规模网络需求的市场中，由于价格和通用性的问题，这样的服务器并没有很强的优势。

4) 基于 PC 和工作站的并行视频服务器簇。这种方案使用高速互联网络，连接多台标准的 PC 或工作站，通过运行一定的软件来完成服务器的功能。在不同实现方法中，可能将一个功能分布到多个物理节点上，或用一个物理节点实现几个逻辑节点的功能。这种结构对于基于计算机的应用具有很好的可扩展性，但在提高 I/O 带宽等方面依然存在问题。

SeaChange 提出的 Broadcast MediaCluster 属于典型的并行视频服务器簇方案，如图 9-20 所示，它将多个相同配置的通用 PC 服务器使用其专利的 SEANET 互联起来，对用户提供一个服务能力和容错能力更强的体系。SEANET 是一个全相联的互联网络，每个节点与其余任一节点之间均有一个全双工链路。通过 SEANET，节点可以访问服务器簇中任意节点存储的数据，每个节目在服务器簇中只存在一份拷贝，实现了各个节点存储设备的完全共享。整个系统的存储能力和 I/O 吞吐量随着节点数增加而线性增加，具有较好的可扩展性。

(2) 分布式层次结构服务器。分布式系统的设计思想是将视频服务器的功能分布到网络中去。这样，对于单机型的视频服务器设计中可能出现的瓶颈，例如输入/输出负载能力和存储能力，就可以分布到网络中。因此即使使用低端计算机作为服务器网络的节点，也能达到很高的服务水平，为小型甚至中型网络提供服务，且价格很有竞争力。

以加州大学伯克利分校提出的分布式服务器为例，系统中有一个站点管理服务器，管理整个系统中的节目内容。数据全部存放于三级存储设备（光盘库或者磁带库）中，由档案服务器控制，在点播节目时，相应的数据复制到视频文件服务器的磁盘中。其中，视频文件服务器可以采用前面提到的几种单机型服务器中的任何一种。如图 9-21 所示。

采用分布式的视频服务器结构，对每个视频文件服务器的性能要求不是很高，因此可以将大量廉价的服务器结合起来，通过合理的控制与调度，以达到高性能服务器的功能。分布式视频服务器具有良好的性能/价格比，而且很容易进行扩展。

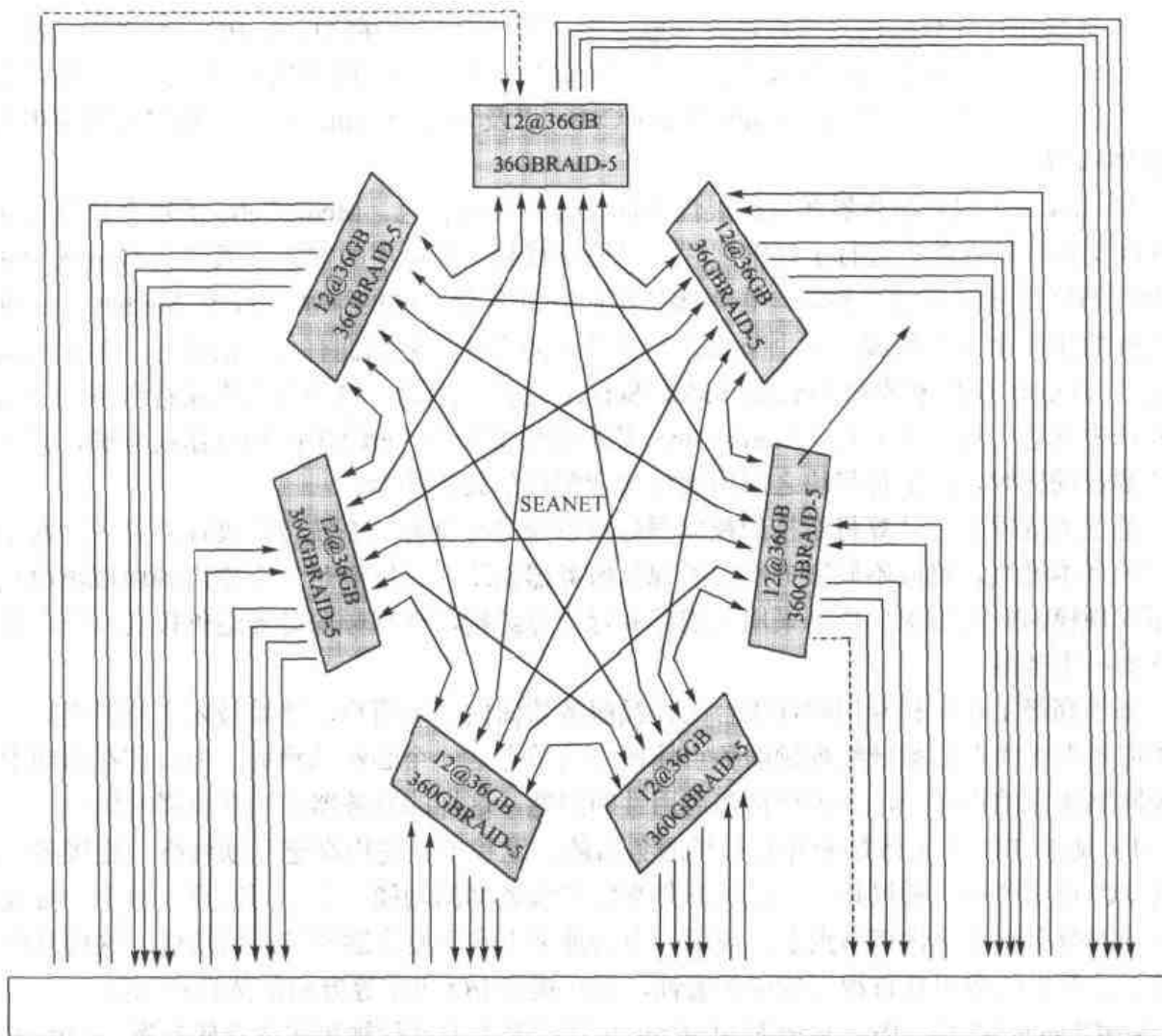


图 9-20 节点的 SeaChange Broadcast MediaCluster 体系结构

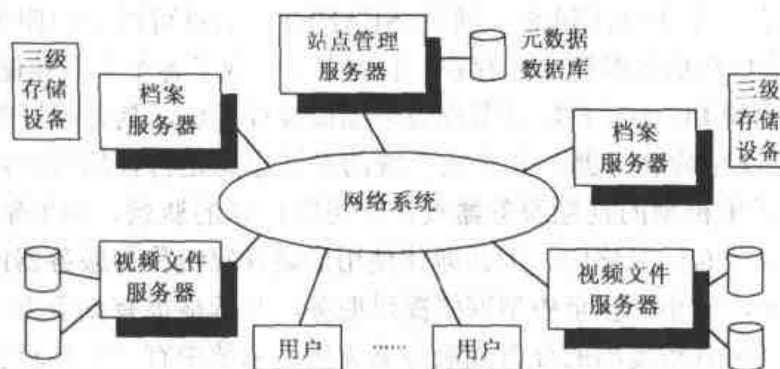


图 9-21 分布式视频服务器体系结构

我们以低端计算机作为视频文件服务器，考察典型的分布式视频服务器的服务能力。每个单独的节点可以支持 50 个 MPEG-1 的视频流，而网络系统中采用 10 个节点作为视频文件服务器，这样网络中的可以支持的用户数目也可以到达 500 个节点，这已经是中等规模的网络系统了。同时这个系统的服务器节点由于采用非专用设备，所以还可以附加其他功能，

例如作为 Internet 的网关服务器等。

实现分布式视频服务器的关键是实现协调各节点的关系并对多节点进行管理。例如档案服务器上的数据传输到哪个视频文件服务器上去？在什么时间传输？如何优化视频文件服务器之间的负载平衡关系？这些问题的不同解决方案对整个系统性能的影响非常大，如果设计优秀的调度算法，整个系统的效率可能显著增加。由于这种系统能力提高不是以增加硬件成本为代价，所以研究者和生产厂商都很关注新的算法的研究。

单纯的分布式服务器也有其缺点。由于视频文件服务器和档案服务器都在同一网络系统中，在服务用户增加的情况下，最先达到瓶颈的将是网络的处理能力。以 1000M 以太网为例，整个网段最大的数据传输率为 1000Mbit/s，但是这仅仅能够满足将 700 路 MPEG-1 的视频流，而这个估算还是在没有考虑从档案服务器传输到视频文件服务器上的网络传输消耗。因此单纯分布式视频服务器系统不利于扩展到大型的网络中去。解决这个问题的办法是，根据网络的具体分布状况，将档案服务器和视频文件服务器放置在不同的网络层次中，建立一种有层次结构的网络体系。另外，在分布式视频服务器系统中，同一节目可能被复制到不同存储节点中，造成存储空间浪费，并且这种浪费随着存储节点的增加而加剧。由于存储子系统的成本可能占到整个系统的一半以上，这种浪费是不可接受的。一些智能的数据复制策略被提出来解决这个问题。而且，服务节点数增加同时提高了系统的管理复杂度和运营成本，因此分布式视频服务器系统的可扩展性也是有限的。

我们在对视频服务器的研究过程中提出一种层次结构设计，它建立在分布式基础上。如图 9-22 所示，直接服务用户的视频文件服务器是层次结构的最底层，每个视频文件服务器可服务一个网段内的所有用户，同时视频文件服务器还连接到与档案服务器相连的骨干网段上。每个视频服务器内的用户形成一个用户组，不同的用户组之间没有直接相连的通路。也就是说，每个用户组独享一个网段的全部网络资源，由于一个网段内的用户数目有限，这样就保障了网络资源不出现瓶颈。在视频文件服务器中可存储一定节目，由于其存储能力的限制，具体存储的方案可根据统计结果而选择热点节目，也可根据用户预约存储预约节目。档案服务器居于系统层次的高层，它是更大的数据存储库，它根据视频文件服务器的需求向视频文件服务器传输节目，而并不直接服务用户。

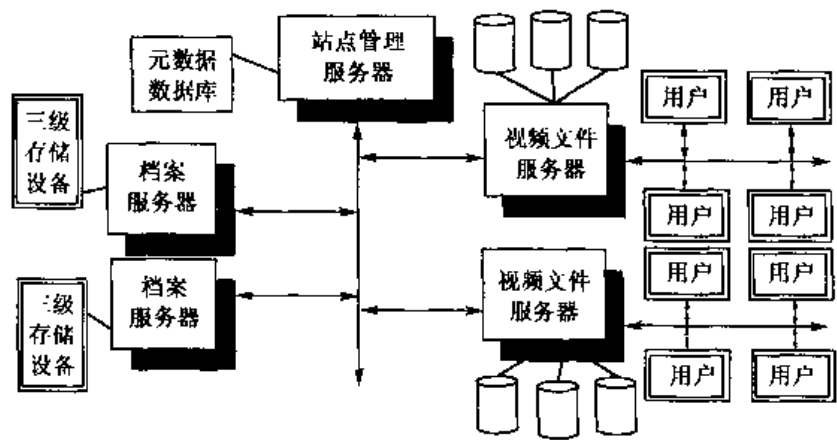


图 9-22 分布式层次结构视频服务器体系结构

简单的两级层次的视频服务器体系可以应用到旅店、企业等中型网络中，多层次则可设

计大型乃至城域结构的视频服务器。如果为每个住宅楼内设置一个视频文件服务器, 数个或数十个住宅楼形成小区设置一级档案服务器, 在数个小区之上再设置更高层次的档案服务器。多层次的建立形成巨大的网络体系, 在这个网络体系中则可以存储海量数据。

分布式和层次结构都是在视频服务器发展过程中提出的新设计思路, 加州大学伯克利分校最早提出分布式视频服务器的体系结构, 多种低端的视频服务器产品采用了这种体系结构, 如 Microsoft NetShow 和 Theater Cisco IP/TV。

### 3. 典型的视频服务器

视频服务器这个概念从理论上提出已经有很长时间, 但是真正的视频服务器产品近几年才进入市场。各家公司推出的产品因为面向领域不同, 因此各种产品性能不尽相同, 下面是一些典型的视频服务器产品。

(1) nCUBE 公司的 MediaCUBE 视频服务器。MediaCUBE 视频服务器采用大规模并行处理体系结构, 通过基本模块的互连共同完成视频服务器的功能。根据市场需求, nCUBE 将不同数目的基本功能模块进行互连, 组成了不同服务能力的视频服务器系列, 满足从小的旅店点播电视系统到公众电视点播的不同规模的应用需求。NCUBE 曾经对 MediaCUBE 做过 5120 个视频流负载下, 无差错运行 72 小时的测试。

MediaCUBE 视频服务器在可扩展性方面的设计十分出色。MediaCUBE 的每个模块设计都对 I/O 资源和处理能力进行了仔细地配置, 十分适合视频点播应用的需求。通过互连这些基本模块, 可以在系统扩展时保证各种资源的平衡, 不会让任何一个环节成为瓶颈。每一个 MediaCUBE 基本功能模块由两个节点构成, 每个节点是一个独立的实体, 它们拥有各自的内存, 独立地运行各自的 Transit 操作系统和 Oracle Video Server 软件。因此, 每个节点在执行任务时不会影响系统中的其他任何一个节点。MediaCUBE 的基本模块之间采用 hyperCUBE 拓扑结构互连, 在增加模块数量的同时, 增加了各模块之间的通信带宽和通信处理能力。

MediaCUBE 的核心是媒体服务器模块 (MSM), 它提供事务处理能力、内存、存储设备 I/O 和网络输出接口。通过增加媒体服务器模块, 视频服务器的存储设备访问、网络输出带宽、存储设备 I/O 到网络接口的互连带宽以及处理能力可以均衡地得到扩展。MediaCUBE 的媒体服务器模块有两种类型, 分别提供对 ATM 和以太网的接口。以 ATM 媒体服务器模块 (ASM) 为例, 它包括两个 nCUBE3 处理器, 两个 OC-3ATM 接口和两个 wide-UltraSCSI 接口。每一个 nCUBE3 处理器拥有各自的 128MB 内存。一个 ATM 媒体服务器模块可以同时为网络提供 90 个 MPEG 视频数据流。

MediaCUBE 的另一个关键模块是内部 SCSI 模块。内部 SCSI 模块 (SSM) 包括两个 nCUBE3 处理器, 每个处理器独立拥有 256MB 内存和 wide-UltraSCSI 接口。SSM 模块主要用于扩展服务器视频文件存储容量和管理能力。

MediaCUBE 的磁盘子系统采用 RAID 技术, 并通过服务器的 hyperCUBE 方式相互连接在一起, 能够提供高带宽、低延迟的磁盘访问。MediaCUBE 还利用 Oracle Video Server 软件, 将视频文件分块存储, 增强了文件并发访问的能力。

(2) Starlight 公司的 StarWorks 视频服务器。Starlight 公司推出的视频点播系统的主要产品为 StarWorks 视频服务器软件。StarWorks 系列产品适合较小规模的视频点播需求。例如, 它的 StarWorks-150Mbps 版本可以支持 100 个 MPEG-1 视频流或 25 个 MPEG-II 视

频流。

StarWorks 支持客户/服务器的模式，将工业标准的服务器平台变为高性能的多媒体服务器，管理视频、音频的发送和存储，可以应用于 Windows NT 和 Unix 平台，支持单 CPU 和多 CPU 系统。客户端支持流行的桌面系统，如 DOS、Windows 和 Macintosh 等。

Starlight 的最大特色在于它的网络协议设计。Starlight 专门为多媒体数据的网络传输设计了传输速率控制协议（MTP），它能控制、测量网络上视频数据的流量，使得一个网段上获得的视频流量大量提高，充分利用网络的带宽，保证网络上高质量的视频传输。MTP 与现有的网络协议（如 TCP/IP、IPX）兼容，它可以运行在现有的共享和交换以太网、快速以太网或 ATM 网络环境中。StarWorks 通过 Starlight 的专有 StarStream 技术来保证网络带宽的有效分配。根据用户对带宽需求的不同而强制分配给用户不同的带宽，提供不同的服务质量。

StarWorks 使用 StreamingRAID 技术进行视频数据的存储管理。此技术可以将视频数据存储多个驱动器或扇区中，通过优化数据放置和磁盘调度，提高磁盘 I/O 的数据并发访问能力。当使用像 Windows NT 这样标准的文件系统时，StarWorks 自适应的 I/O 技术可以管理磁盘访问优先权和缓冲区，提高视频文件的并发访问能力。利用 Starlight 的 StarCenter 视频服务器管理软件，可以根据预计的传输量将媒体流文件在多个视频服务器中进行复制，并跟踪其使用率。

（3）Oracle 公司的 OVS（Oracle Video Server）点播电视系统。OVS 是 Oracle 公司开发的基于软件实现的点播电视系统。它提供了包括视频文件的存储和管理，视频流的建立和发送，客户与服务器的网络通信，视频流的接收、回放等内容的一整套解决方案，同时，它提供了大量的应用程序编程接口和控件，成为一个交互式多媒体系统的集成、开发平台。OVS 可以运行于多种操作系统和硬件平台（包括 Intel/WinNT、Sun、DEC、HP、SGI、nCUBE 等），满足各种规模的点播电视应用需求。

OVS 点播电视系统完全遵从 Oracle 的网络计算体系结构（NCA），包括三个部分：客户端、应用服务器、数据服务器。

OVS 的数据服务器部分用于对视频文件的管理。它控制应用程序对视频文件的访问，管理视频文件在磁盘上的分布，完成从逻辑节目内容到文件物理存储的映射。例如，用户请求播放一部电视节目中间的一个片断，则数据服务器查询数据库，获得这一段视频数据的物理存储位置，然后把对数据的物理存储分布的描述作为一个结构返回给视频服务器，视频服务器根据此结构直接访问文件系统。Oracle 的数据库技术非常出色，它与视频服务器系统的结合也成为 OVS 的一个突出特点。

Oracle 点播电视系统的核心是 Oracle 视频服务器，它构成点播电视系统的应用服务器部分。视频服务器接收和处理用户请求，将对应的视频流发送给用户设备。Oracle 视频服务器包含以下部分：

- Oracle 媒体网络（OMN）。OMN 是 OVS 系统中对网络层的抽象描述，视频服务器、视频客户通过它进行通信而无需考虑底层具体的物理网络。
- 媒体数据存储（MDS）。MDS 是专门为视频数据的存储和实时访问开发的一套文件系统，是 Oracle 视频服务器的专利技术。MDS 将视频文件分块，把这些视频文件块分布到不同的磁盘中，并充分利用 SCSI 技术的特点，实现多个磁盘数据的并行访问，

提高了服务器对存储数据的访问带宽和访问速度。同时, MDS 用软件实现了 RAID 保护技术, 提高了存储数据的可靠性。

- 视频泵。视频泵接收视频流服务部件的请求, 从 MDS 读取视频数据发送给网络用户。
- 视频流服务。视频流服务部分处理用户的请求, 它收集资源信息, 决定是否接收用户的请求。视频流服务部分还负责与数据服务器通信, 确定要发送的视频流数据的存储位置。

(4) Microsoft 和 Net Theater 视频服务器。Net Theater 是 Microsoft 提供的一个媒体点播系统, 这是一个纯软件解决方案。Net Theater 系统中包含 Title Server(目录服务器)、Content Server(内容服务器)两种。Title Server 提供目录服务, 用户通过该服务器获取节目信息, 并且协调 Content Server 工作。Content Server 存储多媒体节目, 向用户实时传输节目数据流。Net Theater 的网络接口可以是 ATM 或者 10M/100M Ethernet, 系统结构如图 9-23 所示。

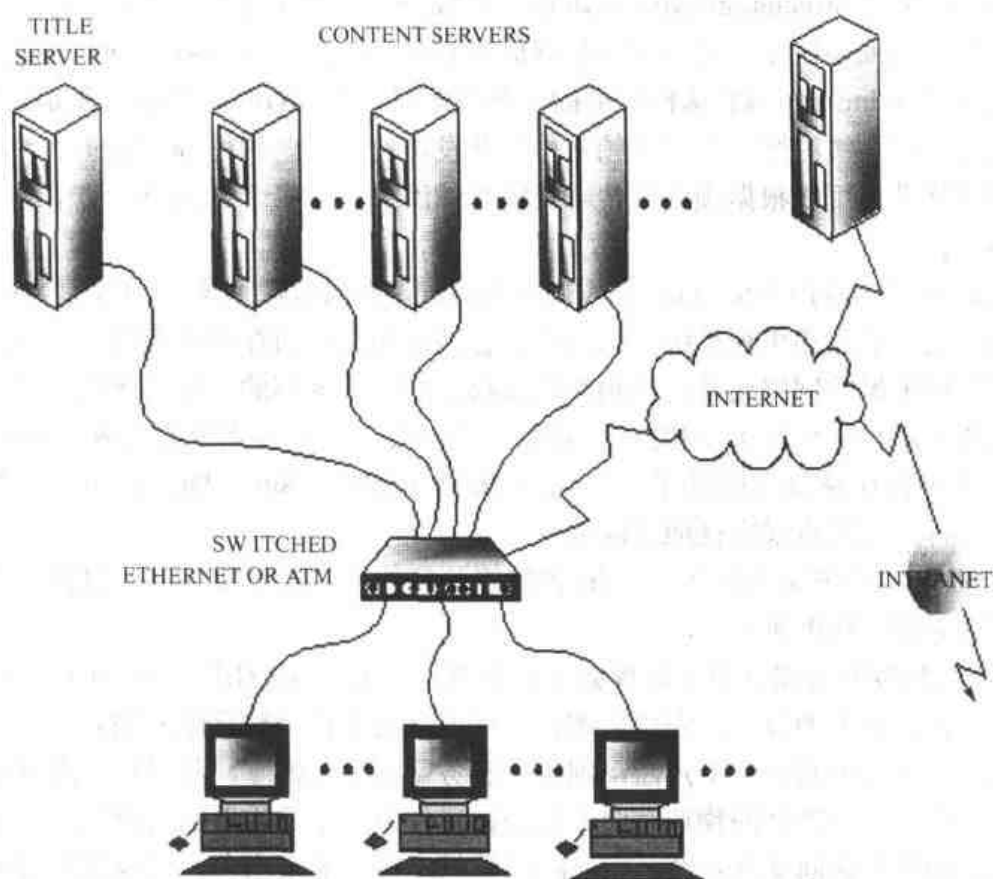


图 9-23 Microsoft 公司的 Net Theater 视频服务器

Microsoft Net Theater 上是一种分布式媒体点播系统, 但是它最多只能够容纳 12 个 Content Server, 限制了系统规模; 其次它要求 Content Server 的主机必须硬件配置一致, 否则整个系统会降低以适应性能最低的 Content Server。

(5) SGI Challenge & MediaBase。SGI 的 Challenge 视频服务器的特点是对图像视频的后期制作管理提供了平台。其优势是作为视频库支持多用户对视频流的并发访问和协同工作。由于 SGI 特长在于其图像处理能力, 因此 Challenge 并不是专为点播使用的视频服务器。



SGI 公司视频服务器 MediaBase 捆绑于它们原有的 UNIX 服务器。SGI 公司对运行在服务器上的 IRIX 操作系统作了一些修改,以满足媒体点播系统的实时性要求。

MediaBase 是集中式媒体点播系统。它们将软件和硬件捆绑在一起,无形中提高了造价。其次在达到系统性能极限时,为了提高系统性能,服务提供者必须淘汰现有的主机,购买速度更快、存储容量更大的主机系统,造成投资浪费。而且按照现在的硬件现状,最强大的集中式媒体点播系统所支持的并发用户数目也是有限的。

(6) MediaCenter UltraSPARC。SUN 公司推出的 MediaCenter UltraSPARC 是一种高性能、低价格、可扩展的系统,适用于通过网络或广播系统进行点播服务,它也是一种中等规模的视频服务器。MediaCenter UltraSPARC 在 SUN 公司原有的高性能 UltraSPARC 服务器以及 Solaris 操作系统的基础上进行扩充与增强,从而达到视频服务器所需要的功能。它的存储系统 MultiPack 采用 20MB 的 Fast-Wide SCSI-2 的接口,使用 RAID-4 结构;在网络接口上,MediaCenter 采用快速以太网和 ATM 两种网络接口。MediaCenter UltraSPARC 服务器的存储容量为 50GB,最大可以扩展到 100GB,即能够存储 124 小时的 MPEG-1 格式的视频数据。服务器的网络接口采用 Fast Ethernet 或者 155Mbit/s 的 ATM,所以一台最多可以支持 67 个 MPEG-1 的数据流或者 25 个 MPEG-II 的数据流。

(7) V-CACHE。FVC 的 V-CACHE 是使用专用硬件平台的小型视频服务器。它的硬件单元采用嵌入式控制器作为整个单元的处理单元,运行其经过简化和优化的操作系统,控制整个单元的处理功能。作为视频服务器,V-CACHE 以 SCSI5 RAID 硬盘阵列提供高容错性和系统可用性,以 OC-3 155Mb/s SCSI-ATM 结构的接口连接高速 ATM 网络。从服务能力上讲,FVC 的 V-CACHE 现在可以支持 50 个 MPEG-1 视频流的同时播放,其存储能力为 50~100 小时的视频节目。

(8) ADS。VideoActive 公司为有线电视网络提供视频点播的解决方案,其视频服务器称作 ADS (ALL Demand Server)。ADS 的核心部件是数字视频引擎 (DVE),每个 DVE 内有一个 RISC 处理器,这个高性能处理器允许 DVE 管理一组 Ultra SCSI-3 盘,每个盘上又存储数个电影节目的数据;DVE 内还有一个高速的视频输送部件,把大量复合在一起的压缩视频流转换成 155Mbit/s 的 ATM 流,再通过 OC3 光纤传至外部的解分设备。ADS 的主机通过 PCI 总线与 DVE 卡相连,并通过以太网把服务器与其他管理计算机相连。主机可选用多种操作系统,例如标准 Windows NT 等,便于不同用户的使用。每个 ADS 最多能控制 12 个 DVE,系统可提供近 1680Mbit/s I/O 输出,即 1000 个 MPEG-1 视频流,这样的服务能力可满足大中规模的应用需求。

(9) MediaStream。HP 的视频服务器称为 MediaStream,它由 MediaStream 广播服务器、MediaStream 磁盘存储器和相应的管理软件组成。磁盘存储器使用 RAID 磁盘阵列存储 50 小时左右的视频数据;广播服务器负责将视频数据通过广播的方式传输给用户,其网络接口为 ATM。它也是一种中型的视频服务器。

(10) DECAIpha8000。DCE 的视频服务器基于其 Alpha8000 系列服务器。该服务器支持多 CPU,其内存、I/O 和总线都是工业级的水平,系统总线吞吐量为 1.26Gbit/s。作为视频服务器,其存储结构是 RAID5 磁盘阵列,对磁盘扇区进行优化保证数据高速存取;网络接口为 SONET 或者 ADSL。

(11) MediaHawk。Concurrent Computer 公司的 MediaHawk 是一种基于高级工作站的



视频服务器。它以专用操作系统 PowerMAX OS 为控制平台。通过 CPU 槽的结构安装多 CPU 来实现系统对处理器的需求。MediaHawk 的存储系统可基于 IDE、SCSI 或者 FC-AL 结构，网络接口可为 10M、100M 以太网或 ATM 网络接口。典型配置下，MediaHawk 可支持 1000 个 MPEG-1 的视频流点播需求。

(12) 国内公司的媒体点播系统。随着媒体点播系统得到越来越广泛的应用，国内一些软件厂商也从事媒体点播系统的研究与开发，并且出现了一些比较成熟的产品，同样这些产品一般是简单的分布式媒体点播系统，还没有考虑分层次的结构。图 9-24 和图 9-25 表示了两个国内公司开发的媒体点播系统结构。

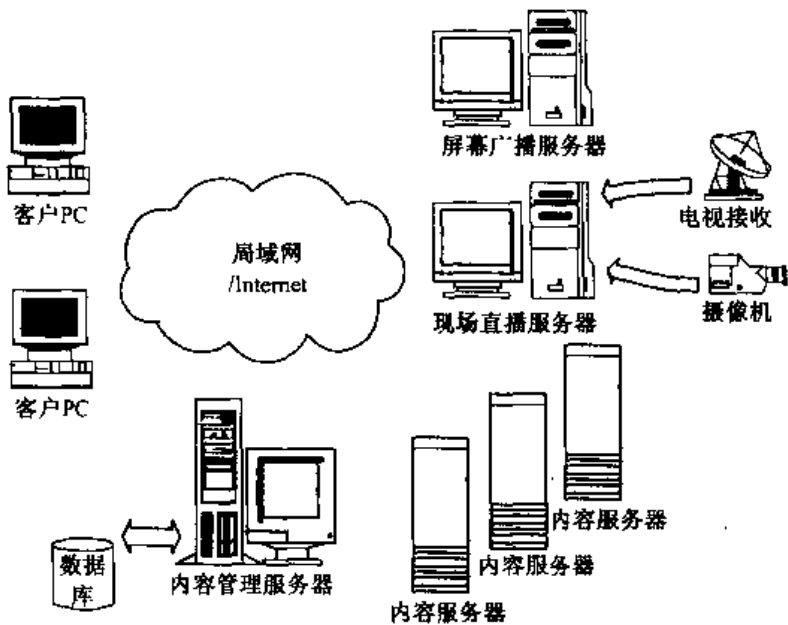


图 9-24 五矿腾龙 IPTV 系统

世纪鼎点公司的鼎点天源视频点播系统有三个服务器程序，这三个程序可以在一台服务器上运行，适合于小规模应用。如果这三个服务程序分布到多个服务器上运行，分担服务器的负载，就适合于大规模的应用。

4. 视频服务器研究的关键问题

视频服务器同普通服务器在功能上的差别巨大：普通服务器面向计算，因此其研究主要集中在计算性能、数据可靠性等问题，而视频服务器则面向资源，其主要技术问题也围绕资源。有效提供大量实时的数据，涉及到对视频服务器外存储设备、内存储设备、存储设备 I/O、网络 I/O 和 CPU 运算等多种资源的协调工作。

(1) 视频服务器的存储问题。即使是小型视频服务器也不可能仅依靠硬盘来存储大量视频节目，而大容量存储设备的访存速度不能适应点播需要。被点播的节目在视频服务器上必须首先调度到硬盘上，再传输给用户，这涉及到节目在硬盘和其他外存储设备上的放置策略。用户对节目点播的分布规律影响放置策略。调查发现，用户对不同节目的收视分布差异很大，热门和冷门节目的收视率遵循一定的规律，这个规律称为 Zipf 分布规律。依照规律，一个有 200 个节目可供点播的视频服务器，最热门节目吸引的观众数目占总观众数目的 21.7%；而前 10 个热门节目的观众占 56.67%，前 20 个热门节目的观众占 72%。也就是说，大部分

观众局限在少数节目上。因此可按节目热门程度的不同，将其放置在硬盘或其他外存储设备上。大部分用户的点播从硬盘得到直接服务，少部分用户点播冷门节目则等待一定时间，视频服务器将节目从外存储设备转到硬盘上再服务用户，这就形成了视频服务器的层次存储结构模型。

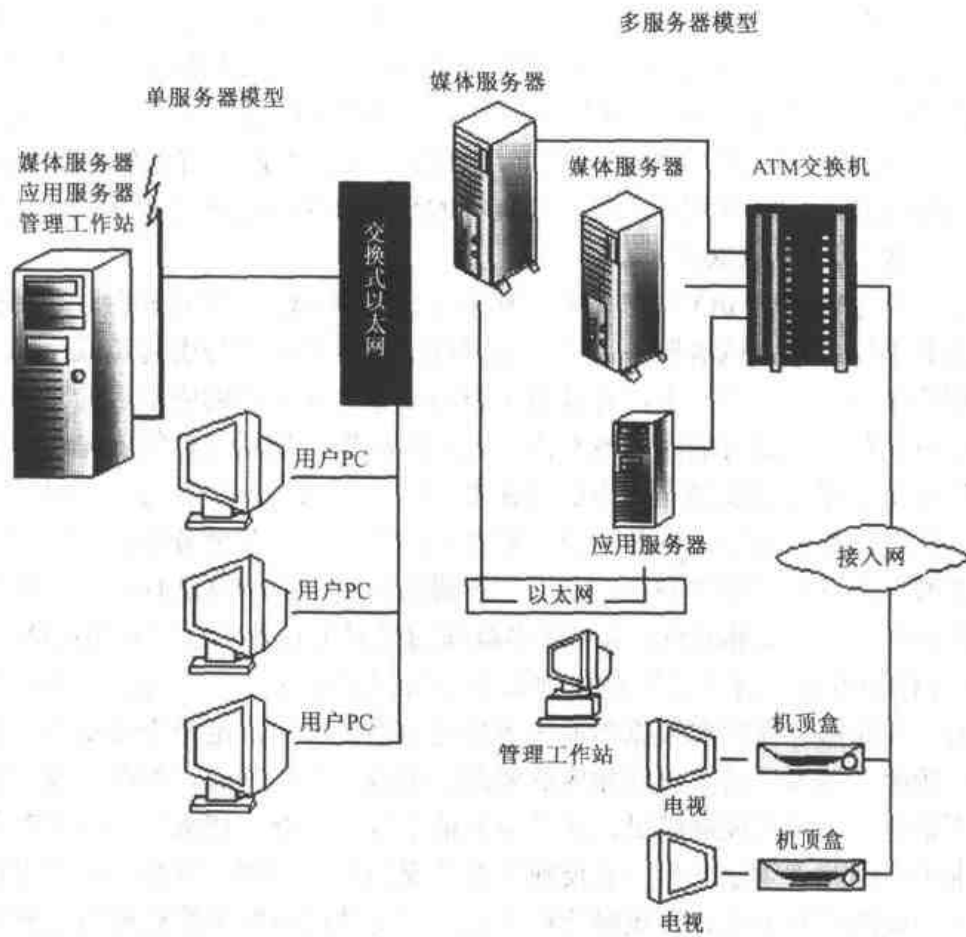


图 9-25 世纪鼎点公司的鼎点天源视频点播系统

实用层次存储结构模型还存在很多问题。例如，热门（冷门）节目的确定尚无有效的策略，观众动态点播节目，视频服务器可对一段时间内的点播进行统计分析，还可引入一定智能预测算法判断热门节目，然而目前并没有出现很好的算法。此外，在确定热门节目后，节目在外存储设备和硬盘间的换入换出的时机与操作流程也有待研究，这个过程占用了系统大量 I/O 资源而可能与系统服务冲突，因此要合理调度此类资源。层次结构存储模型管理结构的设计是影响视频服务器服务质量的关键因素。

(2) 存储设备 I/O。存储设备 I/O 反映了视频服务器同时传输的流数目，从而决定了视频服务器服务能力。磁盘调度和数据分块放置均是解决这个问题的策略。

视频和音频数据的播放需要保证其实时特性，即需要将每个时间片断内播放的数据从磁盘读取并发送给播放器，需要保证播放过程中播放器的缓冲区不出现上溢（overflow）和下溢（underflow）的情况。这和普通文件的访问方式有很大的区别。

为了保证连续媒体数据的实时磁盘访问，需要结合连续媒体周期性、连续性和结构化的特点，将数据以合适的块大小分布到磁盘的各个区域，并以最佳的顺序将数据从磁盘读出。

从硬盘上读取数据,涉及到磁头移动、磁盘旋转、数据读取等许多操作。不同的点播要求可能涉及到同一磁盘上不同数据块在相同时间区间内被读出。在一个限定的时间内,磁头要移动到每个数据块的位置后读取数据,其时间消耗取决于磁头移动速度、磁盘旋转速度和数据块的大小。一般而言,磁盘的旋转速度恒定,读取数据的速率也是固定的,那么减少时间消耗的关键在于如何减少磁头移动。一些研究者提出了不少的磁头移动策略,例如将节目数据按照某种算法组织并放置到硬盘上以减少磁头移动。目前比较成熟的算法是磁盘区域算法,它将磁盘分成若干区域,节目数据按照一定的规则放置在这些区域内,在某个时间段内,磁头只在同一个区域内移动以减少移动时间。这种算法在理论上有效解决了磁盘调度的问题,但是在具体设计上,区域划分、数据放置等的最优算法还待研究和探讨,同时该算法初始延迟时间长、难于进行 VCR 操作。

数据分块策略也同存储 I/O 密切相关。基于充分利用 I/O 带宽的理由,节目数据被分成小数据块放置到不同的硬盘或者服务器上。未分块的节目数据容易引起负载不平衡现象。例如当大多数用户集中在一个节目上,存放该节目的磁盘必须同时响应很多的点播请求而导致忙状态,而此时其他磁盘则可能因为没有点播请求而闲置,使系统总的磁盘 I/O 能力没有充分的利用。解决负载平衡问题的一种方法是在多个磁盘上保持多个备份,不过这种方法过于昂贵。有效方法是将数据分块,将数据块分布到不同的磁盘,甚至分布到不同的计算机上,这样每个点播服务都分配到很多磁盘,从而平衡磁盘负载,这种算法称为分块和交错算法。目前尚无成熟高效分块和交错算法,RAID 多盘阵列采取了有效的近似分块策略。

(3) 许可控制算法。许可控制是视频服务器的技术难题之一。当有一个点播请求到达视频服务器时,视频服务器需要根据当前服务情况与系统容量决定是否接受该请求。一旦请求被接受,视频服务器为该请求分配相应的资源,形成一个视频流。然而,视频服务器的资源有限,并不是每个请求都能够满足,这就涉及请求许可问题。请求许可算法首先要能够监控当前资源水平,因此必须建立起一套反映资源状况的数学模型。现有请求许可控制主要分为基于确定性的策略和基于统计性策略两种方式。确定性的策略主要根据服务器的性能,在理论上推导最坏情况下可以支持的用户数目,从而决定是否接受更多的请求。而统计性的策略使用的数据则基于观察到的服务器性能和过去的服务状况来推测是否能对新的请求进行服务。确定性控制能够对用户的服务质量提供绝对的保证,但是由于其考虑的是最坏的情况,浪费了视频服务器的系统资源。统计性控制能够使资源得到充分的利用,但是只能以一定的概率条件来满足用户服务的实时要求,因此还必须考虑其性能增强的问题。同时,基于统计性的许可控制策略通常需要进行大量的运算,需要耗费更多的时间。

(4) VCR 操作。VCR 功能包括“暂停”、“恢复”、“快进”、“快退”等操作,VCR 功能是实现节目内点播电视的基础,视频服务器响应 VCR 要求是一个比较复杂的工作。

“暂停”、“恢复”是基本操作。如果在准点播电视的基础上实现 VCR 功能,服务器为响应用户的“暂停”操作可以简单的停止用户对数据的接收,但是为响应“恢复”操作,在准点播电视中流的时间安排确定,服务器不可能再建立一条正好适合该用户的视频流,该用户只有被并入一条近似的流,这样对用户而言,观看的节目就可能不连续。

为实现“快进”和“快退”的操作,现有的方法主要有两种,它们都有局限性:一种是在用户方实现“快进”和“快退”操作。将快进或者快退时间段内的数据全都传给用户,用户的解码设备从数据中实现快进、快退操作。这种方法的问题在于数据传输量过大。假设用

户想从影片片头“快进”到影片片尾，那么在很短的时间内，要将几乎整个影片数据传输给用户，这对服务器和网络的要求过于苛刻。另一种是在服务器上实现“快进”和“快退”操作。这种方法在服务器上解码数据，采取选择传输关键帧的方法，将一部分的视频帧传输给用户。这种方法的代价是，服务器必须消耗 CPU 资源解码数据，这加重了服务器的负担；同时用户在“快进”、“快退”操作中也不能看到每一幅画面。并且如果视频数据是 MPEG 这样的可变比特流，那么也会加重网络的负担，因为“快进”、“快退”只能传输压缩率较小的 I 帧，在相同播放帧率的前提下，只传输 I 帧要比平时的传输数据量大 5-10 倍。以上两种方法都不能良好的实现“快进”和“快退”的操作，对这类操作的有效支持是目前视频服务器研究领域的一个难点问题。

(5) 流传输问题。连续媒体流的传输区别于普通文件数据的传输。普通数据文件的传输过程通常只关注数据的传输无差错，传输的速度尽可能短。而媒体流的传输最重要的是保证数据的时间相关性，数据必须在特定的时间间隔内以特定的大小传输，对于少量的数据丢失是可以忍耐的。

为了保证媒体流数据的时间相关性，通常采用无连接的数据报形式发送，同时利用实时传输和控制协议 (RTP/RTCP) 对数据进行封装和发送，以保证传输数据的顺序并监控传输质量以便于动态地调整。

如何有效地封装媒体流数据是一个研究重点，目前，对于 ITUT 的 H 系列的视频流和 MPEG-1、MPEG-2 系统流数据的 RTP 封装算法研究较为成熟，而对于 MPEG-4 和其他的编码格式的媒体数据流封装正在进行。同时，在掌握了网络传输状况后如何动态的调整媒体流的发送也是一个热点问题。

(6) 流调度算法。基于有效利用资源的考虑，视频服务器常常利用网络组播或者广播技术为多个用户服务，也就是说多个用户共享同一条视频流。但在这种合并的情况下，一旦用户进行 VCR 功能操作，就必须脱离这个视频流。此时，如果没有合适视频流可以加入的话，就必须由视频服务器再创建一个新的流。但是，创建新流收到许可控制的限制，因此就可能出现无法实现 VCR 功能操作的现象。这个问题是研究合并算法的一个大障碍。

(7) 磁盘缓存策略。在连续媒体系统中可以采用缓存技术将一部分经常被访问的数据暂时保存在缓冲区中，后继请求再次访问相同数据时，可能发生缓存命中，因此减少磁盘 IO，提高系统性能，为了提高缓存命中率，在内存空间有限的前提下，提出了多种缓存替换算法。间隔缓存 (Interval Caching) 是一种较新的缓存技术，它能够有效复用磁盘带宽，明显提高视频服务器性能。当多个请求访问相同节目时，间隔缓存技术缓存连续视频流，保证后续视频流从缓存中访问数据块，避免磁盘访问，节省了磁盘带宽资源，可以服务其它更多的用户请求。基于间隔缓存技术包括趟缓存 (Run Caching)、细粒度缓存替换算法、粗粒度缓存替换算法，基于对象的缓存算法和替换算法，以及为间隔缓存建立的数学模型等。大部分间隔缓存的现有工作基于 CBR 视频，针对广泛应用的 VBR 视频，有一定局限性。

(8) 加密问题。当许多的用户共享网络带宽时，在系统什么环节对数据加密就成为一个难题。如果在视频服务器存储前加密 (预加密)，那么一旦掌握密码，人们就可以很容易偷看节目；如果在数据传输到用户的时候加密 (动态加密)，视频服务器则丧失主动权；如果在传输中对不同用户分别加密，则要解决用户共享带宽的方式，同时还要考虑加密代价是否消耗视频服务器或系统很多的资源。

对上述所有问题的有效解决都将很大程度上提高视频服务器服务能力或服务质量，解决这些问题则是研究人员的重任。

## 9.2.4 交互式电视机顶盒的结构和设计

所谓机顶盒 STB (Set\_top\_box)，最初是指放在电视机顶端或内部的一种终端装置，可以由观众通过遥控器按键控制有线电视节目的接收与转换，其功能和电视技术的发展相适应。近年来随着数字电视技术、通信技术的发展和 Internet 的迅速普及，机顶盒的作用从单一的解密收费发展成为集数字电视解压缩、Internet 浏览、解密收费、交互控制和家庭网关为一体的数字化装置，这一领域已经成为计算机和消费类电子工业发展的重点。

### 1. 机顶盒的功能

(1) 概述。机顶盒 STB 是消费电子 (consumer electronics) 领域内的一个名词，它指的是一种与电视机接口，并提供附加服务的设备。目前所谓的机顶盒有四种类型：传统的机顶盒通过电缆接收数据，第二种类型使用电话线，第三种类型应用于 DBS (Direct-Broadcast Satellite) 服务，第四种机顶盒就是一个 Video-CD 播放器。我们这里所说的机顶盒主要指前两种。

1) 机顶盒的产生和发展。最初，机顶盒的引进是为了弥补电视调谐器能力的不足。如果调谐器屏蔽不够好，无线广播的电视信号将混入电缆传输的有线电视信号中，影响图像质量。于是出现了一个类似于黑盒子的设备，用它接收电缆信号，进行外差振荡，把信号的频率移到广播所不占用的频带上。这便是最早的机顶盒。

第二代机顶盒增加了可编址的能力，即每个机顶盒对于视频服务器具有唯一的地址。这样，用户从有线电视广播的信号中选择与自己地址相匹配的数据接收。通过这种方法，视频服务器可以对用户有选择地发送视频信号。

目前的机顶盒已进入了交互式的一代，它无论在解码能力、显示效果还是交互性方面，都远远超过了前两代。

2) 机顶盒的分类。根据不同的分析角度，机顶盒可以有以下几种分类方法：

(a) 模拟型机顶盒和数字型机顶盒。模拟型机顶盒接收和处理模拟信号。这意味着数据解码和数/模转换的功能需要在传输之前由视频服务器进行。另外，对网络的传输能力也有很大的要求。数字型机顶盒所接收的信号是压缩数字视频信号。因此，机顶盒本身需要具有解压缩和模拟电视信号编码的能力，其价格要大大高于模拟型机顶盒。

采用模拟型机顶盒的系统把费用由机顶盒转向了服务器，由于机顶盒的数量很多，所以系统整体的价格降低了。但是，服务器解码的能力有限，所以限制了用户的数量。目前的交互式电视系统使用的基本上是数字型机顶盒。

(b) 非智能型机顶盒和智能型机顶盒。非智能型机顶盒对于用户的交互请求，只负责将其传送到服务器，所有交互服务的处理都在服务器中完成。由于服务器具有大容量的虚存（物理磁盘），应用程序的开发可以按照常规的方式进行。智能型机顶盒本身执行一部分应用程序，对于高度交互的服务（如交互式视频游戏）可以提供立即的响应。智能型机顶盒需要大容量的内存和较高的处理能力。

关于机顶盒引入智能的问题，有一种看法（见 95Computer Graphics 关于机顶盒的一个讨论会的记录）认为，机顶盒中重要的是具有一个良好的界面，而智能化的问题应留给计算

机来解决。

(c) 基于计算机的机顶盒和基于电视的机顶盒。机顶盒的设计可以基于一台工作站或 PC 的硬件，并在计算机的屏幕上进行输出；也可以单独设计硬件，连接到电视机上，并用遥控器进行控制。

但是两种类型的机顶盒今后几年内不可能完全融合，因为电视和计算机在使用上具有以下几点基本差异：

- 电视一般具有很大的屏幕，而同样大小屏幕的监视器对计算机来说十分昂贵。
- 电视在收看时习惯上是被动的，而计算机的使用具有高度的交互性。
- 电视很容易使用，而计算机具有复杂的界面，需要经过专门学习。
- 电视只需要用遥控器控制，而计算机带有许多接口设备，如键盘、鼠标、游戏杆等。
- 电视在观看时一般距离很远，所以显示的文字信息很少，而计算机一般离得很近，显示大量的细节信息。

电视的使用主要是为了娱乐，而计算机主要用来提供信息、办理业务等。所以，选用哪种类型的机顶盒需要根据系统提供的主要类型而定。

(2) 机顶盒的功能和设计目标。机顶盒作为用户前端设备，它是用户控制交互式电视的接口。所以除了能够提供一定质量的音频与视频输出之外，还应该完全支持系统交互。

1) 机顶盒的功能需求。一个交互式电视机顶盒，应该具有以下几项功能：

- 提供与网络的接口。这是机顶盒的一个最基本的功能。机顶盒从网络中选择所需信息，并把用户的命令经过返回通道送给服务器。
- 音频与视频的解码。机顶盒需要能够提供高质量的视频和音频输出。用户希望视频质量能够达到广播级：NTSC 格式的分辨率为  $720 \times 486$  像素，而 HDTV 格式则需要  $1920 \times 1080$  像素。此外，用户希望能提供 CD 质量的音频，即以 44.1kHz 或更高的速率对每个声道进行 16 位采样。
- 用户界面和图形控制。提供一个友好的界面，使用户更容易地使用。机顶盒应该具有电子节目导游 EPG (Electronic Program Guide) 功能，通过交互选择，使用户方便地从众多服务中挑选自己所需要的。
- 外围设备控制。最普通的外围控制设备是红外线遥控器。为了支持视频游戏，机顶盒可以连接游戏操纵杆。此外，机顶盒还可以连接 CD-ROM、录像机、打印机、键盘等，提供更多的功能。
- 安全与权限管理。安全与权限管理是机顶盒与一般计算机的一个关键区别。与一般计算机所不同的是，机顶盒的基本功能是提供付费服务，这保证了视频提供者获得一定的报酬，并对用户合理地收费。

机顶盒还可以包括其他扩展功能。这些功能可以通过可选件的形式实现，由用户根据需要，选择购买和使用。

2) 机顶盒结构的设计目标。由于机顶盒的设计与网络基础设施的建设密切相关，所以在考虑机顶盒结构的设计目标时，应注意以下几个方面：

- 长期的使用。机顶盒设计时必需考虑能使用 8 年以上。
- 廉价。支持低廉的价格和一个最小的功能集合。
- 可扩展性。很容易增添新的部件，提供新的服务。

- 兼容性。与多种服务网络的软硬件平台相适应。
- 数据独立性。与数据类型无关，以便支持将来出现的新的服务类型。

总之，机顶盒的设计必须体现开放性原则，使机顶盒并不停留于现有技术水平上，从而提供了最大的灵活性和最多的功能。

## 2. 机顶盒的硬件结构

数字接收机顶盒一般由四大子系统组成，即控制子系统、数字处理子系统、用户/扩展接口子系统和网络接口子系统，图 9-26 为 STB 的硬件功能框图。

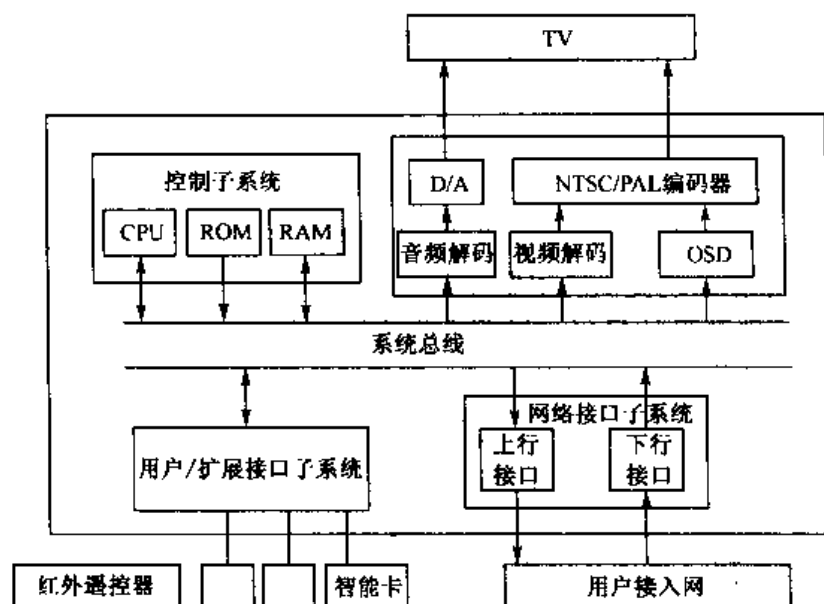


图 9-26 STB 硬件功能框图

(1) 控制子系统。由 CPU、ROM、DRAM 等组成，CPU 是 STB 的核心，负责各子系统的初始化以及各部分的协调工作。ROM 中包含自举代码和基本操作系统程序，一般在系统启动时由 CPU 控制将 ROM 中的操作系统核和二进制服务程序和数据装入到 DRAM 的相应位置，然后开始运行。

(2) 数字处理子系统。包括视频处理模块，音频处理模块和图形处理模块。

1) 视频处理模块。对 MPEG-2 传输流进行解复用并对压缩视频流进行解码，通过从控制系统接收指示信息，从网络接口子系统接收压缩视频数据流。开放式视频解压缩器装入适当的基于某种算法的微码（如 JPEG、H.261、MPEG-1、MPEG-2）进行解压缩。对 SDTV 模式，进行 ML@MP 解码，对 HDTV 模式进行 HL@MP 解码，此模块的输出与 OSD(On Screen Display) 系统产生的图形混合，经 NTSC/PAL/SECAM 编码器传送给 TV 显示。

2) 音频处理模块。包括解码与合成两个部分，压缩编码声音数据经过音频解码器解码，输出至 D/A 转换器，得到 TV 的伴音信号。

3) 图形处理模块。提供类似计算机的复杂图形甚至三维图形，需要系统具有强大的二维、三维图形加速芯片，而且能够将图形有视频信号进行混合叠加显示。平台图形界面的要求由服务种类和服务复杂程度决定。

(3) 网络接口子系统。由网络通信芯片执行通信和网络接口功能，如调制解调、加密

与解密、权限管理等，支持外向网络接口如 Cable、Modem、ADSL 和 HFC 等接入网的全双工通信。

(4) 用户接口及扩展接口子系统。用户接口及扩展接口子系统可设置 IEEE1394、HomePNA、SWAP 等家电网络接口，可使用户采用红外遥控器、游戏控制器、智能卡、解密卡等外设进行交互式操作。

3. 机顶盒的软件结构设计

机顶盒的软件设计可以采用一个层次型的结构，其优点在于使底层的硬件对上层软件透明，增加和替换硬件不用修改高层的软件，上层软件修改时不必了解硬件的结构。这样升级和扩展起来十分方便。图 9-27 是机顶盒的软件层次结构。

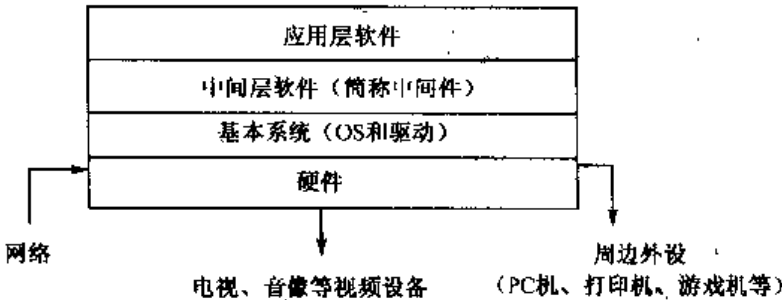


图 9-27 机顶盒的软件层次结构

STB 各软件层次性能见表 9-10。

表 9-10 机顶盒各层次性能

应用软件	各种多媒体网络应用，数字节目处理
中间件	应用协议处理 用户/业务处理 机密管理 费用管理
OS (基本软件)	存储容量小 实时 多任务

(1) 实时的嵌入操作系统。通信领域的多媒体应用处理对象通常是周期性的实时数据流，对于操作系统的实时性要求要高一些。终端的硬件实现途径有多种形式，其 RTOS 应考虑对多种类型常用处理器支持的可能，在微内核的下面采用硬件抽象层（HAL）可以隐藏处理器和外设的具体形式，提高操作系统的可移植能力。价格低、性能优、鲁棒性和可靠性强也是通常要求满足的。

(2) 中间件层。目前趋势是将应用层软件建立在中间件（middleware）的基础上，这几年 OpenTV、PowerTV 和 NCI 等成为中间件的主流，这些中间件的共同特点是支持 DAVIC 建议中的大多数功能，Sun 公司的 JAVATV API 也是为此开发的。

采用这种明确的层次结构使得整个系统的软件结构清晰、具有较好的扩展性并可以提供除电视节目收视之外的更多的其它功能，同时便于应用程序的移植，但缺点也是明显的，最终用户必须为软件支付较高的费用，而在这些软件所提供的功能中有很多并非用户所需要的。

(3) 应用程序层。数字电视解压缩、Internet 浏览、解密收费、交互控制和家庭网关等等。

4. 机顶盒工作过程

机顶盒设计的一个重要原则是开放式结构，以便为用户提供更多的服务，采用功能模块化结构，可使系统软件具有灵活的可扩充性。



数字节目接收功能的机顶盒一般实现的功能模块组成可用图 9-28 表示。

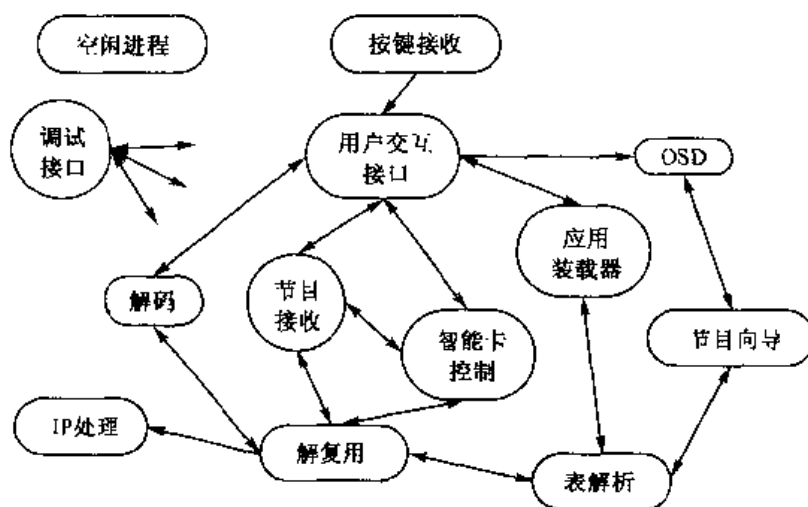


图 9-28 数字接收机顶盒一般实现功能模块及关系

数字接收机顶盒所处理的实时事件基本上都是数据输入事件，这些数据来自卫星、远红外线遥控器 IR、智能卡、串口等等。典型机顶盒提供许多不同的功能，这些功能可以划分为一些标准模块的集合。

数字电视接收机顶盒的工作过程一般有以下五个：

1) 先由主控单元完成对整个系统的控制，包括系统初始化、控制信号接收及系统复位等等。

2) 当机顶盒开始工作时，它接收（从卫星或有线）符合 MPEG 标准的数字压缩电视节目，进行 QAM（有线）或 QPSK（卫星）解调，将传来的数字电视节目流处理转化为解复用所需的 MPEG-2 系统码流。

3) 多路解复用完成对输入的 MPEG-2 系统码流进行解复用、解包的处理功能。

目前的传送系统使用了由 MPEG-2 定义的具有固定长度的传送流（TS）分组方式。一个传送流分组包含一个或多个节目，每个节目有一个或多个复用在一起的基本流构成。传送层支持的基本码流包括视频、音频、数据及系统控制信息等，解复用模块从系统码流中分离出用户需要的某路节目，并将解包后的压缩视/音频流数据交给解码单元处理，同时将数据信息（如私有数据，EPG 节目信息，IP 信息包等等）视情况相应处理。

在系统信息中，PSI（Program Specific Information）和 SI（Service Information）用来传递各种相关的 MPEG-2 系统参数（PSI）和拥有那些必要参数的 DVB 特定信息（SI）。传送流中的传送包以一个 4 字节的前缀开始，它包含一个 13 位的分组包标示 PID（Package Identifier），通过节目特定信息表 PSI，PID 可以识别传送分组包中所包含的数据内容。

在传送流中运载的 PSI 表有 4 种：节目关联表（PAT）、节目映射表（PMT）、网络信息表（NIT）、条件访问表（CAT）。这些表中包含解多路和显示节目所需的条件。

- 节目映射表 PMT。制定了哪些 PID 和哪些基本流是相互关联的，从而构成一个节目。同时它还给出携带节目参考时钟（PCR）传送流的 PID。
- 节目关联表 PAT。作为 PID=0 的 TS 包的有效数据进行传送，而将节目映射表 PMT

作为有效数据进行传输的 TS 包的 PID 是在节目关联表 PAT 中给出的。主控单元 CPU 首先利用 PID=0 的节目关联表 PAT，识别出携带节目映射表（PMT）的位流的 PID，然后从节目映射表（PMT）中获得构成该节目的（视频、音频用数据等）PES 基本流的 PID 值，从而对所选节目进行解码。

DVB 服务信息表 SI 中，用于提供节目信息的表有 NIT、SDT 和 EIT：

- 网络信息表（NIT）。提供关于可用网络 and 不同传送流的物理传输参数的信息（如 QPSK/QMA，频率等）。
- 服务信息描述表（SDT）。提供所有与网络提供服务有关的服务信息（传输流，服务名称，服务类型和服务体供名）。
- 事件信息表（EIT）。提供由 EPG 表格显示的各种信息，如开始时间，持续时长，节目类别和节目的文字描述信息等。

所有这些 PSI 和 SI 信息都能在解复用时加以利用，继而可以构造如电子节目表格 EPG 这样的节目预报信息来使用户更为方便的选择观看。

4) 解码部分接收解复用后的压缩音、视频码流，按照 MPEG 编码格式分别对视频码流和音频码流解码。解码器通过对起始码检测，使音频、视频和系统同步。

将解压缩的视频信息传给 NTSC/PAL 编码器，然后显在 TV 上；将解压后的音频信息传给 D/A 转换，然后输出到麦克风；有时能提供视频图像各特殊显示功能，如缩放，静止图像等；提供音频的静音，非静音功能等。

5) 面板或遥控器接收及 OSD 屏上显示。接收通过遥控器按键或控制面板按键来的用户控制信息，对其进行相应处理（如频道切换命令由解复用处理、静音命令由解码器处理等等），并以 OSD（On Screen Display）屏上显示的形式显示机顶盒的工作状态、各诊断信息和用户命令等。

上面的这五个功能模块相互协作就可以完成一个数字接收机顶盒最基本的功能了，此外，对于经过加密处理的节目流还有条件接受处理模块处理付费接收；有的机顶盒支持对下载文件的打印输出功能，具有 IEEE1284 接口以及相应的打印处理模块；有的机顶盒支持动态下载功能，其内有应用装载处理模块；这样的功能模块还有很多，视不同的功能需求而有所不同。

## 9.3 小结

根据通信节点的数量，视频会议系统可分为点对点视频会议系统和多点视频会议系统。视频会议系统主要由视频会议终端、多点控制器、信道（网络）及控制管理软件组成。在 20 世纪 80 年代，ITU 就专门成立了一个小组研究视频会议，从那时起建立了一系列的视频会议系统的建议和标准。

综合业务多媒体通信终端可以广泛用于多媒体通信、交互式分布多媒体系统，特别是各种计算机支持的协同工作系统（CSCW）的通用设备，是目前世界各国大力研究的热点课题。由于各种技术发展较快，因此要求总体结构设计要考虑通用性和可扩展性。

MCU 是视频会议系统中的关键设备，它的作用是图像、语音、数据信号进行切换，而且是对数据流进行切换，并不象电话交换机对模拟信号进行切换。

交互式电视和分布式多媒体技术有很多相同之处。交互电视系统要适应广泛的应用领域,其许多功能分散在各子系统内。这些功能有共性功能和特殊功能,未来应用还将定义和增加新的功能。从系统的操作、完整性和开发方面看,交互电视系统核心功能应包括位传输、会话、访问控制、导航与节目选择、应用启动、媒体同步链、应用控制、表现控制、使用数据(用户使用系统信息)和用户概况等。从交互电视应用与服务角度上看,交互电视系统应具有一些公共功能。

通用 VOD 系统主要的组成部分有视频服务器、编码器/网络路由单元、用户请求计算机和记账计算机、位于用户处的机顶盒。

## 9.4 习题

1. 视频会议系统的关键技术是\_\_\_\_\_。  
(A) 视频会议系统的标准。  
(B) 多点控制单元 MCU。  
(C) 视频会议终端。  
(D) 视频会议系统的安全保密。
2. 下面哪一种说法是不正确的? \_\_\_\_\_。  
(A) 视频会议系统是一种分布式多媒体信息管理系统。  
(B) 视频会议系统是一种集中式多媒体信息管理系统。  
(C) 视频会议系统的需求是多样化的。  
(D) 视频会议系统是一个复杂的计算机网络系统。
3. 多点控制单元的作用是什么?
4. 视频会议系统最著名的标准是\_\_\_\_\_。  
(A) H.261 和 H.263  
(B) H.320 和 T.120  
(C) G.723 和 G.728  
(D) G.722 和 T.127
5. 在视频会议系统中为什么需要 QoS?
6. 视频会议系统设立了哪些访问权限?  
(1) 超级、优先。  
(2) 一般、作废。  
(3) 排队、优先。  
(4) 超级、中断。  
(A) (1) (2) (B) (3) (4)  
(C) (1) (4) (D) 全部

## 9.5 参考答案

1. (B)

2. (B)
3. (略)
4. (B)
5. (略)
6. (A)